THERMAL CONNECTORS FOR SELF-RECONFIGURING MODULAR ROBOTS: SOLUTIONS AND APPLICATIONS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Jonas Neubert

May 2014

THERMAL CONNECTORS FOR SELF-RECONFIGURING MODULAR ROBOTS: SOLUTIONS AND APPLICATIONS

Jonas Neubert, Ph.D.

Cornell University 2014

For self-reconfiguring modular robots the function to autonomously form and break connections between neighboring modules is a defining trait. Connectors must be strong, reversible, transmit power and signal, consume little power, and should be small and light enough for several to fit into each robot module. To make the modular robot system as versatile as possible, the connector should also be genderless, rotation agnostic, and tolerant to misalignment and positioning errors. Traditionally, connectors add significant mechanical complexity to each module.

This thesis introduces two hardware components that implement different aspects of the modular robot connection process in small lightweight packages that contain no moving parts: The thermorheological valve is a device for directing fluid flow in order to selectively attract robot modules in a stochastic fluidic self-assembly system. Suitable thermorheological fluids are identified, the mechanical and thermal design of the valve is optimized, and demonstrations of mesoscale thermorheological valves are presented. The soldering connector is a connection mechanism for modular self-reconfiguring robots that forms solder joints by heating a low melting point alloy. The connector is mechanically strong, reversible, electrically conductive, only requires power when connecting and disconnecting, and supports ten to hundreds of connection cycles. Compared to other connection methods for modular robots the soldering connector, which is contained on a single printed circuit board, is very low in complexity, weight, size, and cost. Several designs of the connector are presented and evaluated and general guide-

lines for application specific connector designs are provided.

The thermorheological valve and soldering connector are then applied in the design of two modular robotic systems. The first proposed application is a fluidic self-assembly system of one-inch-sized, cube shaped modules that rely on stochastic external actuation to form predefined target shapes. Module prototypes and proof-of-concept experiments for this system are presented. A second application implements a system of 40 cube shaped modules that uses the self-soldering connector for self-reconfiguration. Several demonstrations of interactions that mimic the flow of matter through an ecosystem are shown in experiments and simulation, including growth and self-refinement.

BIOGRAPHICAL SKETCH

Jonas Neubert was born in 1987 in the Erzgebirge region of what was then the German Democratic Republic. After his family emigrated to West Germany shortly before the reunification of the country in 1989, Jonas spent most of his childhood in the rural town of Überlingen.

Thanks to his parents' initiative and generous scholarship support, Jonas later spent six years at two boarding schools, Schule Schloss Salem and Sächsisches Landesgymnasium Sankt Afra. Both schools enforced a broad liberal arts curriculum which led Jonas to pursue interests ranging from the Ancient Greek language to computer programming. A thesis project and internship investigating automation technology in a semiconductor fab helped finally manifest Jonas' primary interest in manufacturing and automation technologies.

Jonas received a Master of Engineering in Mechanical Engineering, a four-year undergraduate degree, from the University of London's Imperial College in 2008. Following an internship in a steel factory after his sophomore year, Jonas questioned his goal of a career in industry and set out to find research opportunities in the field of robotics. Jonas followed Prof. Hod Lipson's invitation to spend the summer at the Cornell Computational Synthesis Lab where he developed Tetrabot, a robot that exploits vibration for locomotion without external moving parts. The day after returning to England, Jonas began the graduate school application process which led him to return to Cornell in 2008.

While attending graduate school at Cornell, Jonas also brought the Ignite event format to Ithaca as a co-founder of Ignite Ithaca, and served two terms as a teaching assistant with the Cornell Prison Education Program at the maximum security correctional facility in Auburn, NY.

For Conny.

ACKNOWLEDGEMENTS

The work presented in this thesis was funded in part by the Defense Advanced Research Projects Agency (DARPA) under grant #W911NF-08-1-0140, the National Science Foundation (NSF) under grant #0735953, and an annual doctoral fellowship by the German Academic Exchange Foundation (DAAD). The content of this document is solely the responsibility of the author and does not necessarily represent the official views of the sponsoring organizations.

My gratitude belongs to:

Hod Lipson for first inviting me to work in his lab as an undergraduate; for playing a central role in forming a vision for my future and the decision to apply to grad school; for providing me with guidance, support, and ingenious research ideas, and creating the unique work environment that is the Creative Machines Lab.

Michael Tolley for being an important source of advice during my first two years at Cornell, and being a great travel companion on several shared conference trips.

Michael Schmidt for sharing many insights into computer science and the local cycling community.

Stephane Constantin, Jeremy Blum, and Rob MacCurdy for their mentorship while I learned electronics the hard way.

Daniel Ly for being a great desk neighbor and the hundreds of insightful conversations.

All the wonderful people, too many to name them all individually, who were my colleagues in the Creative Machines Lab and formerly Computational Synthesis Lab, for sharing their creativity and friendship.

The faculty at Cornell, especially Hadas Kress-Gazit, David Erickson, Ashutosh Saxena, Hadas Ritz, and Pam Staub, who were great teachers and colleagues.

The staff at Cornell, especially Marcia Sawyer, Gabe Terizzi, Patti Wojcik, Craig

Ryan, and Carol Moss, all of whom saved my day countless times.

My ENGRI1170 students for making it fun to be their a teaching assistant.

My prison students for having a thirst for knowledge and gratefulness to their teachers that I have not experienced elsewhere, and for giving me a glimpse into their life that is so different.

My parents and grandparents who support my sister and me unconditionally in every possible way while we seek the education that was not available to them.

Cornelia Scheitz, my love and my best friend, who often has more trust in me than I have in myself.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CAD	Computer Aided Design
DC	Direct Current (e.g. DC motor)
EEPROM	Electrically Erasable Programmable Read-Only Memory
GUI	Graphical User Interface
IC	Integrated Circuit
LED	Light Emitting Diode
LSB	Least Significant Byte
MSB	Most Significant Byte
PCB	Printed Circuit Board
PLC	Programmable Logic Controller
SMA	Shape Memory Alloy
USB	Universal Serial Bus

CHAPTER 1

INTRODUCTION

1.1 Modular Machines

This thesis is about self-reconfiguring modular robots. Self-reconfiguring modular robots are composed of many similar or identical modules that can be actuated in ensemble to change the shape or function of the robot. The oft-cited promise of such systems (for example in [1–6]) is for

- **Versatility:** For a system with n cube shaped modules, each cube face containing a rotation-agnostic connector, there exist $(24)^n$ shape-distinct configurations.[1] When actuated modules are considered, the design space of kinematically distinct configuration grows exponentially with the number of modules, too. Thus, given a sufficiently large number of modules, a reconfigurable modular robot systems would allow for many different arbitrarily complex machines to be "programmed" by re-arranging the same modules.
- **Low Cost:** For modular systems where the number of modules per machine exceeds the number of module types by orders of magnitude, fabrication might benefit from economies of scale. Once the challenge of designing a simple easy to manufacture module is solved, the availability of cheap bulk-produced is possible. The construction of a robot then transforms from a complex integrated design task in a continuous design space into an optimization of the assembly sequence of discrete components, making it more accessible to humans and more

[1] $(cw)^n$ is the trivial upper bound for passive modules with c connectors and w relative orientations between each connector pair [7]. Chen et al. offer a more precise method for computing the number of non-isomorphic configurations for certain types of modular robots that takes into account the degrees of freedom of the modules in [8].

suitable for automated design optimization. While today it seems unlikely that small-scale electromechanical robot modules will one day be produced in bulk and arranged into robots by algorithms, some argue that the similar developments in high-volume production of nanoscale electronics seemed equally unlikely just a few decades ago [9].

- **Robustness:** Modularity is a key concept in design for maintainability [10]. A self-reconfiguring modular robot system could use its capability to change the arrangement of its modules in such a way that partially functional modules are moved to locations in the robot where their missing functionality is not required. In modular system that support exchange of modules with the environment modules can be discarded or replaced with new modules.

These promises are not supported by the current state of research in modular robotics. Kasper Stoy's remarks at the Robotics Science and Systems conference 2005 about current modular robots being "(1) useless, (2) expensive and (3) they break all the time" [11] still largely holds true today in 2013. Most research systems presented to date offer little utility beyond demonstrating self-reconfigurability on a small scale.

As always when technological progress falls behind imagination, science fiction offers motivating examples: The Sandman in the movie *Spiderman 3* is composed of sand grain sized modules that allow for shape and functional reconfiguration. Somewhat closer to the scale of existing current robot modules are those that make up the Replicators in the TV series *Stargate*. Some also cite the T1000 robot in the movie *Terminator 2* as a modular robot, though the shape-shifting alloy material it is composed of appears to be continuous and not made of discrete modules.

One challenge in the implementation of large scale self-reconfiguring modular robots is the construction of a suitable connection method between modules. The avail-

ability of a simple, versatile, strong, repeatable mechanical and electrical connector would have the potential to unlock innovation in many other aspects of modular robot design. The first part of this thesis in Chapters 3 and 4 describes components and strategies for the connection mechanism between modules in self-reconfiguring modular robots.

1.2 Programmable Matter: Modular Robotics & Assembly

When studying the two and a half decades of literature on the design and implementation of self-reconfiguring modular robots one encounters various approaches to the topic: Some early work aims to mimic cellular automata [12–15], others' primary goal is to demonstrate gaits [1, 16–19], and yet others pursue biological phenomena such as self-replication [20, 21].

Another group of self-reconfiguring modular robotic systems is designed with focus on the process of self-reconfiguration [22–26]. The only function of modules in such a system is to participate in the two components of self-reconfiguration: forming and breaking connections between modules. The system does not have a secondary function once a target shape has been reached.[2] In order for the system to be self-reconfiguring, no external manipulation must be involved in the reconfiguration process.

Modular assemblers are machines that assemble passive or active structures from a reservoir of homogeneous or near-homogeneous building blocks, for example microspheres [28], latching tiles [29] and interlocking links [30]. Most recently, MacCurdy et al. developed such a "digital material" with electromechanically active components

[2] Systems that perform self-reconfiguration can perform locomotion by iteratively updating the target shape in a way that results in an aggregate motion from one place to another [27].

which are put in place by an external assembly machine [31]. The output of a modular assembler looks similar to a modular self-reconfiguring robot, but the process of creation is fundamentally different: Self-reconfiguring modular robots do not require external actuation to change state, modular assemblers do.

The boundary between self-assembly and assembly is blurry. Halfway between digital assemblers and self-reconfiguring modular robots there exist systems where the external assembler is of a similar scale as the assembled components, for example truss reconfiguration robots [32]. Externalizing parts of the reconfiguration process might provide solutions to some of the technical challenges of implementing self-reconfiguring modular robots. Chapter 6 proposes an application of the thermorheological valve described Chapter 4 towards a self-assembly system that autonomously controls the assembly process, but relies on random external actuation for module transport. This concept of stochastic assembly is a common mode of assembly at the molecular scale but its implementation at the mesoscale remains challenging [33, 34]. In the context of modular robotics, stochastic assembly has the potential of reducing complexity and size of modules and at the same time it could open the door to massive parallel assembly of structures.

1.3 Machine Metabolism and Ecosystems of Robots

Modularity is a common phenomenon in the natural world. Most fundamentally, all matter is constructed from modules, atoms, of less than 120 known types, the elements. In biology modularity occurs at all scales: At the molecular level metabolic pathways and gene regulation have been claimed to be modular [35], and at ecosystem scale many network interactions between organisms and communities are modular systems.

Modular machines offer the opportunity to emulate concepts observed in nature through engineered systems. The idea to construct modular machines to simulate biological processes including self-reproduction, growth, and speciation was first presented by von Neumann in 1948 [36]. Since then modular biological systems have served as inspiration for many concepts in computing [37]. But despite the potential benefits of ecosystem interactions for autonomous machines, only a few simple robotic implementations of concepts from biological networks exist. This might in part be due to the complexity involved in the design and construction of small electromechanical systems like they are required in modular hardware systems. Hence, the simple components for modular robots described in this thesis could provide a step towards the construction of larger ecosystems of robots.

Chapter 7 proposes and report on first experiments of a robot ecosystem: A system of self-reconfiguring modular robots which can exchange parts with their environment, including other robots, to form an interdependent network of robot organisms. Using experiments and simulations, it is shown that these modular robots can exhibit growth and self-refinement behaviors, which are commonly associated with metabolism in living organisms.

CHAPTER 2

RELATED WORK

2.1 Self-Reconfiguring Modular Robots

2.1.1 Taxonomy of Modular Robots

In its broadest sense, the term *Modular Robot* defines any robotic machine that is composed of distinct functional modules. This is not a useful definition because with it one could argue for any robot be a modular robot. After all, most robots are assembled from discrete functionally independent components such as actuators and microcontrollers. However, there are certain robots whose degree of functional separation is so high and amount of coupling between modules so low that being modular becomes their defining characteristic. Standardization and interchangeability are properties of components of these modular systems [38].

Regularity is a metric orthogonal to modularity that describes how homogeneous a system is. Regularity does not imply modularity, and neither does modularity imply regularity: A modular system can be composed of many fundamentally different module types, or homogeneous consisting of many modules of one type only. Therefore, homogeneity is a measure for the reconfigurability of a system, particularly functional homogeneity and shape homogeneity.

Most robotic systems are tightly integrated systems that lack well-defined interfaces between their functional modules and are neither modular nor reconfigurable. Most consumer products fall under this non-modular category including robotic products from house cleaning robots to personal telepresence robots.

One area of robotics where the term "modular robot" finds use is for robotic arms that can be configured from a selection of modules specific to an application. For example, the German company Schunk advertises their articulated robotic arms which can be configured from an assortment of actuation, gripper, sensing, and link modules, as "modular robotics" [39]. This kind of modular robot has a high degree of modularity but is heterogenous because each module is specific to its position in the arm resulting in the number of modules per arm assembly normally equals the number of module types.

LEGO® Mindstorms construction kits, as well as other robotic toy construction kits, similarly exhibit a high degree of modularity. In addition the LEGO® system has a higher degree of homogeneity with the number of modules per assembly typically exceeding the number of module types.

This thesis focuses on robots whose module count exceeds the module type count by orders of magnitude, that is robots with homogeneous and near-homogeneous module shape and function.

2.1.2 Modular Robots vs Swarm Robots

The terms *modular robot* and *swarm robot* are sometimes used interchangeably. A short excursion into the origin of the latter explains why. During the 1980s cellular automata were a very active are of research and several researchers pursued the goal of cellular robots as a hardware implementation of similar multi-agent behaviors. Beni's 1988 paper [40] formulates a mission statement for cellular robotics, while Fukuda et al. published about the first attempt at constructing a cellular robot in the same year [41]. In a personal retrospective the same Beni describes his dislike of the terms cellular automata and cellular robot and the subsequent introduction of the "buzz word" *swarm* [15]. After

his 1989 paper on *swarm intelligence* [14], the term became common nomenclature for describing simulated systems with very large numbers of agents, even appearing in popular media (for example [42] and [43]). Because of the inherent difficulty in constructing similarly large swarms in hardware, robotics researchers used the more conservative descriptor *modular* for their systems until the mid 2000s when the term *swarm robotics* re-emerged in literature [44].

Historically modular robot and swarm robot were both introduced as synonyms for cellular robot, but the two terms reflect inherently different approaches to the same problem:

Swarm robot describes the *behavior* of the robot. In a swarm the interaction of a large number of relatively simple normally disconnected agents results in a complex collective group behavior [15,44]. Common examples of swarming behaviors are flocking [45] and pheromone-based foraging (used in ant colony optimization) [46]. Members of a swarm can be physically connected or disjoint.

Modularity describes the *morphology* of a robot as consisting of multiple modules that are physically connected. Only the act of forming a physical connection creates a modular robot [4,5]. The modules can be complex or simple and may exhibit collective behaviors.

Because swarming refers to the behavior and modularity to the physical construction of the robot, the two descriptors are not mutually exclusive. Figure 2.1 illustrates the relationship between the two groups of robots. There exist systems that exhibit swarming behavior but whose components are not physically connected, for example Fukuda et al's MARS [47] and Mondada's Khepera [48]. On the other side there exist systems which are modular but do not exhibit swarming behavior, like the aforemen-

tioned robotic toy construction kits and other manually reconfigurable modular robots. Several systems exhibit traits of both: Mondada's Swarm-Bot modules, for example, are normally not connected but carry mechanical connectors for forming temporary connections [49], while the reconfigurable modular CKBot is normally connected but has been demonstrated to be able to reassemble even when broken into disjoint parts [50]. This thesis is primarily concerned with the physical construction of modular robots which is why the term "modular robot" is used throughout, irrespective of the behavior of the system.

2.1.3 Self-Reconfiguration

After delineating the term "modular robots" above, a brief discussion of the use of "self-reconfiguring" is in order to fully define the heading of this section. A system is reconfigurable when its components can be rearranged by an external entity. If the system is capable of changing the arrangement of its components without external input, it is self-reconfiguring[1].

The ability to self-reconfigure is not a binary property of a system. Yim points out that reconfigurability is a continuous scale from purely manual reconfiguration to completely autonomous reconfiguration [38]. Chapter 6 will discuss a system that relies on external energy input by means of stirring a tank for reconfiguration, but is considered self-reconfiguring because the system controls the connection and disconnection steps. The system described in Chapter 7 is designed to be fully autonomous and does not rely on external actuation or manipulation during its self-reconfiguration.

The above considerations are visualized in Figure 2.1 which relates the terms

[1] "Self-reconfiguring" and "self-reconfigurable" are used in literature with approximate equal frequency. "Self-reconfiguring" is used throughout this thesis.

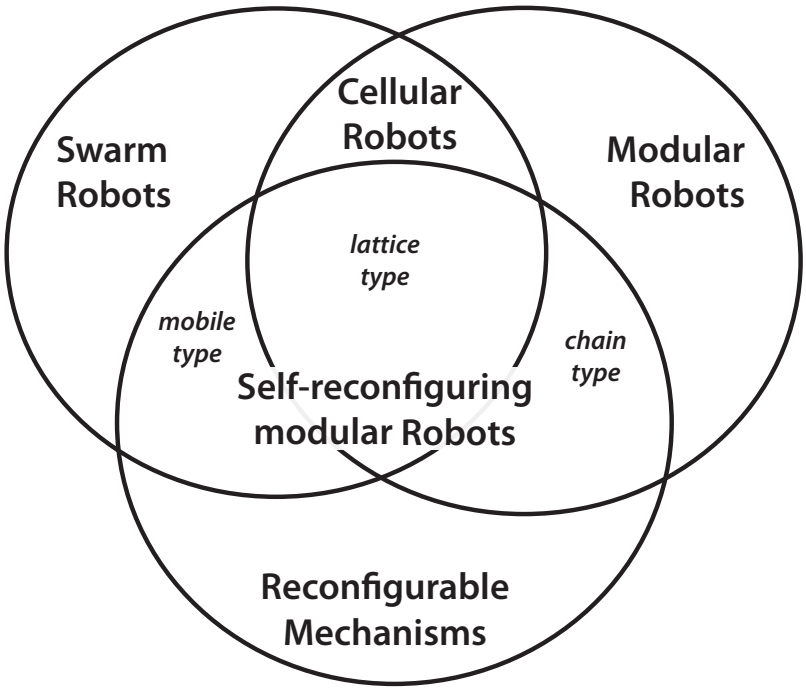


Figure 2.1: Relationship between swarm robots and modular robots. Because swarming refers to the behavior and modularity to the physical construction of the robot, the two descriptors are not mutually exclusive.

swarm robot, *modular robot*, and *reconfigurable robot*, whose intersections are self-reconfiguring modular robots.

2.1.4 Definitions

To enable self-reconfiguration in a modular robot, each module must possess some degree of autonomy. In many cases every module has actuators, sensors, power supply and a controller, making it a robot in its own right. In fact, at least two recent self-reconfiguring modular robot systems are specifically designed such that individual modules can locomote autonomously (i-Mobot [18], SMORES [51]). To avoid confusion when discussing robots made of robots, the following definitions of common terms are

adhered to throughout this thesis:

- The *module* is the atomic unit of a modular robotic system that cannot be non-destructively divided into subcomponents.
- A *meta-module* is groups of $n \geq 2$ modules acting as one logical unit. Such meta-modules may have a smaller number of kinematic constraints than the original module while being n times larger and more complex. This construct can be advantageous for reducing the computational complexity of controlling modular robots as proposed by Prevas et al. [52], Christensen [53], and Brandt et al. [54], among others. In the scope of hardware implementation, meta-modules can be useful when design constraints are such that one module cannot accommodate all intended module functions, but two permanently joined modules can accommodate all intended functions of a module pair.
- The term *robot* is used exclusively for physically connected assemblies of modules actuated together to modify the workspace, for example through locomotion or rearrangement of modules. [2]
- A *system* is a collection of modules that interact but may not be connected to each other at all times. A system can contain one or more robots as well as modules not associated with a robot. Robots in a system can interact such that the number of robots and the number of unassociated modules in the system changes over time, but the total number of modules in a system is constant.

[2] There exists a theoretical edge case of one module performing such actions, which is not encountered in this thesis.

2.1.5 Survey of Modular Self-reconfiguring Robot Systems

The progress of the field of self-reconfiguring modular robotics can be understood from the high level considerations and reviews published by Yim et al. in 2002 [4], in 2007 [5], and in 2009 [6]. Yoshida et al. [55] provide a review of the decade-long work on modular robotics at the Japanese National Institute of Advanced Industrial Science and Technology (AIST). The most extensive review of self-reconfiguring robotics to date is the book by Stoy et al. [11]. More recent efforts are the very comprehensive paper by Moubarak and Ben-Tzvi [19] with focus on applications in mobility, and a brief snapshot by Madhevan and Sreekumar [56].

The following paragraphs are a chronological list of selected modular robotic systems that will be referenced again the later chapters of this thesis, each with a short description of their respective distinguishing features and contributions to the hardware design of modular robots. In addition, Table 2.1 provides a tabular overview over the systems mentioned together with a selection of key characteristics. Reading the remainder of this section is not necessary for following the subsequent parts of this thesis; instead it is provided as a resource to which one may refer when other modular robot systems are mentioned later on.

CEBOT (1988) While the authors use the terms *Cellular Robot* and *Dynamically Reconfigurable Robotic System*, CEBOT is commonly referenced as the first physically implemented swarm robot. Inspiration for the construction of CEBOT comes from the goal to implement the concept of cellular automata in a real-world system. The evolution of the heterogeneous 2D system consisting of wheeled autonomous units design through several iterations can be traced through several papers by Fukuda et al. [12, 41, 57–59].

Polypod (1993) Yim's *Polypod* system was designed with the primary goal to demonstrate a novel form of locomotion. Contrary to CEBOT, Polypod's modules cannot move autonomously, but various gaits of locomotion including rolling and quadrupedal were demonstrated with assemblies of modules. Polypod is a three dimensional heterogeneous system with two module types, configured as a chain, and while assemblies of Polypod modules are able to move, individual modules are not. Polypod has connectors designed for autnomously docking, which makes Polypod the first self-reconfiguring modular robot. [1, 38, 60]

Metamorphic (1993) Concurrent to the development of Polypod, the 2D system *Metamorphic* was designed at Caltech with the primary goal of demonstrating self-reconfigurability in a lattice [61]. Metamorphic modules take the form of a 3 dof six-bar linkage which connects to its neighbors through an error tolerant coupling mechanism that allows for module deformation during re-configuration. A second variant of Metamorphic with square tiles whose connectors allow for modules sliding along each others sides is also presented and a small number of modules was demonstrated experimentally for each version. [62].

Fracta (1994) Inspired by Fukuda's work on CEBOT, researchers at AIST in Japan constructed the *Fracta* modular robotic system to translate the cellular nature of biological systems to engineered systems. The goal of emulating traits of (simulated) cellular automata such as self-configuration, self-assembly, and self-repair drove the design towards a two-dimensional homogeneous system with local only communication [63]. Fracta modules, three of which were constructed, consist of a middle plate holding three electromagnets, sandwiched between top and bottom layers that contain permanent magnets. Actuation and connection are conflated: Switching the electromagnets

both turns the center disk, and forms or breaks connections with neighbors. [2, 63]

Micro-Unit (1998) Starting with the Fracta system, the group at AIST continued exploring the design space of modular robotics systems by implementing demonstration systems, at least six of which have been published to date. After the 1.2 kg heavy 125 mm diameter Fractum modules [2], *Micro-Unit* is an attempt to implement the same two-dimensional lattice based concept in a smaller module size. This is achieved by using SMA instead of DC motors as actuator. Similarly to Fracta, Micro-Unit connectors are actuated such that actuation of one module results in the neighboring module changing it's position in the lattice. Actuation is, however, independent from the connection mechanism. While there exists only one module type, modules are arranged at alternating 90° rotations to ensure a male connector always meets a female connector of neighboring cubes. Several versions of Micro-Unit exist, ranging in weight from 80 g to 20 g per 2D module [55, 64, 65]

3D-Unit/3D Fracta (1998) In parallel to working on a miniaturized modular robotic system with the Micro-Unit, Murata et al. at AIST also extended their Fracta concept to three dimensions with *3D-Unit*. This is the first three-dimensional lattice type modular robot. 3D-Units are cube shaped modules with six orthogonal connector arms which can each be rotated independently. Despite using a complex power transmission to require only one DC motor for actuation and connectors, a 3D-Unit weighs 7 kg, has a nominal width of 265 mm, and relies on external power supply and control [66].

CONRO (1998) The *CONRO* system developed at the University of Southern California (USC) also pursues the objectives of module autonomy and miniaturization: Each module has on-board power, IR emitter-receiver pairs are used for inter-module commu-

nication, and on board processors were used to demonstrate various behaviors. Similarly to Polypod, the system is of chain type, but has only one type of module and makes use of several novel (at the time) technologies to shrink the module size. Additionally, publications about CONRO are the first to mention, but not implement the concept of a robot exchanging modules between with the environment, or even the exchange of modules between robots. [16, 67–71].

PolyBot (1998) PolyBot is a successor to Polypod [3] with applications in distributed locomotion. A first version "G1" requires manual connection and disconnection of modules, while a second "G2" version allows for self-reconfiguration of the assemblies [72]. A third generation "G3" adds sensors and improves the overall design for manufacturability and robustness [4, 73, 74]. One of the first papers dedicated specifically to designing connectors for self-reconfiguring modular robots is written in the context of PolyBot [75].

Molecule (1998) The Molecule system by Kotay et al. introduces the concept of metamodules: Two permanently joint "atoms" form one "molecule". In a lattice, each metamodule takes up two square lattice cells. Actuation of Molecule's is by changing the relative configuration of the atoms inside a molecule as well as actuation of some external connectors. [76]

Crystalline (1999) Crystalline is presented by the same group at Dartmouth as Molecule in the prior year. Crystalline is the first system to use linear actuation instead of rotation. Each module can extend its connection surfaces outwards by one module length, effectively letting one module push or pull itself by one lattice position relative to its neighbors. Unlike the Molecule, Crystalline can be closely packed in 2D

space. [77–79]

MTRAN (2000) Based on their previous experiences with Fracta, Micro-Unit, and 3D-Unit, Murata et al. conclude that the construction of a 3D lattice type system is a significantly harder mechanical design problem than chain type systems or 2D systems [80, 81]. Their proposed solution is the Modular Transformer (*MTRAN*) whose modules are meta-modules consisting of two nearly identical halves which are permanently connected by an actuated arm. The internal actuators allow the module change the relative position of the two cells between three configurations. Where 3D-Unit had six actuated connectors per lattice cell, allowing for any arbitrary pair of modules to arbitrarily change their relative configurations, MTRAN only provides passive connections between modules and only allows a subset of possible configuration changes between the two halves of the meta-module. [80]

MTRAN II (2002) After MTRAN I implemented a successful tradeoff between functionality restrictions and manufacturability, its successor MTRAN II contained several detail improvements that turn the concept into a feature complete modular robotic system: On board batteries offer autonomy of modules, inter-module communication through the connectors allow for untethered operation, and several actuator and control improvements yield sufficient precision to complete self-reconfiguration scenarios. A total of 20 modules were constructed and a large number of behaviors from various gaits to self-reconfiguration were demonstrated. Several publications exist that present work in progress, [82] and [83] are the most up to date hardware design references.

Telecubes (2002) Through ingenious use of materials, *Telecubes* is able to extend the concept of linear actuation first introduced by Crystalline to three dimensions. Sadly, it

appears that only two Telecube modules were ever constructed.

Chobie-II (2003) After all prior systems are designed for locomotion or self-reconfiguration as an end in itself, Chobie-II is the first system whose expressed goal is the dynamic reconfiguration into passive mechanical structures. The primary novelty in module design is the use of keyed sliding connectors allowing for modules to translate along the surface of neighboring modules without ever releasing the connection. [84,85].

ATRON (2003) ATRON arguably is the first system constructed with the expressed goal of manufacturing 100 or more modules, shifting the focus to questions surrounding scalability and reliability. This is prominently reflected in the design choice of only having one degree of freedom per module. As a result, even simple lattice reconfiguration operations require assemblies of several robots, such as the four-component metamodules described in [54]. One hundred ATRON modules were constructed [54] and since then the system has been used as a demonstration for many algorithms covering reconfiguration and locomotion. [86, 87]

Molecubes I (2005) Molecubes was built specifically with the application of self-replication in mind. Molecubes too have only one degree of freedom but six connectors per module, but differ from other systems in placing their dof to rotate about the long axis of the cube, allowing for a larger reach to be implemented with the same number of modules, when compared to ATRON, for example. *Molecubes II* is an open source derivative of Molecubes I that is manually reconfigurable. [20] [21]

Roombots (2010) A *Roombot* module kinematically resembles a metamodule of two Molecubes with an addition rotational degree of freedom at their connection. This

removes the constraint that partitions 3D space into two distinct voxel-grids between which modules cannot transfer, by which all systems composed of cube shaped modules with one degree of freedom are affected. With the relatively larger Roombots, this greater space of possible reconfigurations is applied in the context of reconfigurable furniture [88].

ModRED (2010) ModRED is a chain type system with two connectors per module, presented in the context of applications in space exploration. The system differentiates itself from other chain type modular robots by having a combination of rotational and prismatic degrees of freedom per module. [89–92]

CoSMO (2013) The *Collective Self-reconfigurable Modular Organism* is a recent system that deviates from the trend towards less complex modules and instead consists of general purpose self-reconfiguring robot modules suitable for many self-assembly concepts. Each module contains no less than seven processing cores, the communication system is a switched Ethernet, and modules transfer up to 200 W of power between each other. This shows how advances in electronics allow for the integration of more and more capable modules. However, the impressive spec sheet comes at the cost of large size, each module is 10.5 cm^3, weight, 1.25 kg per module, and, presumably, cost [93].

2.1.6 Survey of Manually Reconfigurable Modular Robot Systems

Superbot (2006) Shen et al. attempt to apply the concept of self-reconfiguring robots to unstructured environments, particularly lunar exploration, with *Superbot*. The Superbot module is similar in design to MTRAN-II with an added roll degree of freedom.

Because an active connecter has never been implemented, Superbot is a manually reconfigurable modular robot.

Molecubes II (2007) The *Open Source Molecubes* apply the mechanical layout of the original self-reconfiguring *Molecubes I* to a general purpose manually reconfigurable modular robotics platform [94]. An "extended version" contains gripper, wheel, camera, and battery modules, in addition to the original actuated and passive modules [95].

CKBot (2008) The *Connector Kinetic roBot* (CKBot) can be considered a successor to Polypod and PolyBot with a focus on simplicity and cost optimization. CKBot is a heterogenous system with module types including two types of actuated module, various wireless communication modules, IR distance sensor module, and a camera module. The system has been used for a number of demonstrations, for example in self-reassembly of a robot after an explosion [50] and the fastest locomotion among modular reconfigurable robots [96].

Vacuubes (2011) *Vacuubes* are not a complete modular robotics system (they lack actuation and control), but a test bed for a vacuum based connection mechanism between modules [97, 98].

Table 2.1: Self-reconfiguring modular robot systems

Name	Affiliation	Year	Ref.	Dim.	Category	dof†	homog.?	Actuation	Size (mm)	Weight	No. built
CEBOT	SU Tokyo	1988	[41]	2D	mobile	N/A	✗	DC motor	180x90x50	1.3kg	—
Metamorphic	JHU	1993	[99]	2D	lattice	3 (3)	✓	DC motor	—	—	2
Polypod	Stanford	1993	[1]	3D	chain	2 (2)	✗	DC motor	—	—	—
Fracta	AIST	1994	[2]	2D	lattice	1 (1)	✓	electromagnet	D:125 H:160	1.2kg	3
Metamorphic2	Caltech	1996	[62]	2D	lattice	2 (2)	✓	DC motor	—	—	2
3D-Unit	AIST	1998	[66]	3D	lattice	6 (6)	✓	DC motor	265x265x265	7kg	2
CONRO	USC ISI	1998	[16]	3D	chain	2 (2)	✓	DC motor	108x25x25	115g	—
Micro-Unit v1	AIST	1998	[100]	2D	lattice	2 (2)	✓	SMA	50x50x50	50g	6
Molecule	Dartmouth	1998	[76]	3D	lattice	4 (4)	✓	DC motor	D:102	3.2kg	1
Vertical	RIKEN	1998	[101]	2D	lattice	2 (2)	✓	DC motor	90x90x90	—	4
Crystalline	Dartmouth	1999	[102]	2D	lattice	1 (1)	✓	DC motor	51x51x178	340g	24
I-Cube	CMU	1999	[103]	3D	lattice	3 (3)	✗	DC motor	85x37x18	205g	—
Micro-Unit v2	AIST	1999	[65]	2D	lattice	2 (2)	✓	SMA	20x20x30	15g	2
MTRAN	AIST	2000	[80]	3D	hybrid	2 (2)	✓	DC motor	66x132x66	440g	—
PolyBot	PARC	2000	[3]	3D	chain	1 (1)	✓	DC motor	50x50x50	200g	56
Telecubes	PARC	2002	[104]	3D	lattice	6 (6)	✓	DC motor	60x60x60	—	—
Chobie-II	TITech	2003	[26]	2D	lattice	2 (2)	✓	DC motor	80x80x75	500g	7
Gear-Unit	Ryukyus	2003	[105]	2D	lattice	1 (1)	✓	DC motor	D:55	—	3
MTRAN-II	AIST	2003	[82]	3D	hybrid	2 (2)	✓	DC motor	60x120x60	400g	—
Stochastic 2D	Cornell	2004	[22]	2D	lattice	0 (0)	✓	stochastic	60x60	100g	3
ATRON	USD	2005	[86]	3D	hybrid	1 (1)	✓	DC motor	D:110	850g	100
Catoms	CMU	2005	[106]	2D	lattice	0 (0)	✓	N/A	D:44 H:40‡	105g	—
Molecubes	Cornell	2005	[20]	3D	hybrid	1 (1)	✓	DC motor	100x100x100	625g	7
Prog. Parts	U. Wash.	2005	[107]	2D	lattice	0 (0)	✓	stochastic	L:120 H:42	—	—
Stochastic 3D	Cornell	2005	[23]	3D	lattice	0 (0)	✓	stochastic	100x100x100	—	4
XBot	UPenn	2007	[108]	2D	lattice	0 (0)	✓	stochastic	—	—	5

Table continues on next page

Table 2.1 continued

Name	Affiliation	Year	Ref.	Dim.	Category	dof	homog.?	Actuation	Size (mm)	Weight	No. built
MTRAN-III	AIST	2008	[109]	3D	hybrid	2 (2)	✗	DC motor	65x65x130	420g	50
ModRED	U Nebraska	2010	[110]	3D	chain	4 (4)	✓	stepper motor	368x114x119	3.17kg	—
Roombots	EPFL	2010	[88]	3D	hybrid	3 (3)	✗	DC motor	220x110x110	1.4kg	2
M3 Express	JHU	2012	[111]	2D	lattice	3 (3)	✓	DC motor	—	878g	2
SMORES	UPenn	2012	[51]	3D	hybrid	4 (4)	✓	DC motor	100x100x90	—	2
CoSMO	KIT	2013	[93]	3D	hybrid	1 (1)	✓	DC motor	105x105x105	1250g	2
M-Blocks	MIT	2013	[112]	3D	lattice	N/A	✓	flywheel	50x50x50	143g	8

† Degrees of freedom (actuated dof in parentheses) ‡ Estimate from figures. — = Information not found in published literature. ✓= yes. ✗= no.

2.1.7 Recent Commercial Applications

In addition to the work at research institutions that produced the modular robotic systems mentioned above, several attempts to commercialize such systems have been made in recent years.

iMobot The iMobot reconfigurable modular robot system was introduced at the ICRA2010 conference by Ryland et al [18, 118] and subsequently marketed by Barobo, Inc, for professional and research applications. The robot module is similar in design to the MTRAN-II and MTRAN-III systems with the unique feature of two added degrees for continuously turning two connector plates that can also act as wheels. As of October 2013 the iMobot system is no longer marketed.

Cubelets The *Cubelets* system is a heterogenous manually reconfigurable robotic toy manufactured by Modular Robotics, Inc. Sold in packages of 20 or six modules, each Cubelet is a 43 mm side length cube that connects to neighbors with permanent magnets. Sixteen different module types exist with each module serving one specific actuation, sensor, or compute function, but all modules are identical in shape. One unique feature of Cubelets is, that by including modules types for basic computational functions like negation or addition, the shape of a Cubelets robot also defines its behavior. Cubelets are a continuation of Schweickhardt's et al. *roBlocks* system [119, 120]. To date (October 31, 2013) the retail price of a set of 20 Cubelet modules is US$520, and 50,026 Cubelets have been produced [121].

Linkbot Linkbot is a toy robot construction kit sold by iMobot, Inc. In contrast to the heterogenous Molecubes system, there are only two Linkbot modules that differ by

Table 2.2: Reconfigurable modular robot systems

Name	Affiliation	Year	Ref.	Dim.	Category	dof[†]	homog.?	Actuation	Size (mm)	Weight	No. built
SuperBot	USC ISI	2005	[113]	3D	hybrid	3 (3)	✓	DC motor	168x84x84	500g	6
YaMoR	EPFL	2005	[114]	2D	chain	1 (1)	✓	DC motor	—	—	6[‡]
CKBot	UPenn	2006	[115]	3D	chain	1 (1)	✗	DC motor	60 x 60 x 60	138g	—
DoF-Box	EPFL	2007	[116]	3D	chain	1 (1)	✓	DC motor	45x45x135	180g	—
Miche	MIT	2007	[24]	3D	lattice	0 (0)	✓	N/A	45	130g	20[‡]
Molecubes II	Cornell	2007	[94]	3D	hybrid	1 (1)	✗	DC motor	66x66x66	200g	7
Odin	USD	2008	[117]	3D	chain	1 (1)	✗	DC motor	D:35 L:60	—	—
iMobot	UC Davis	2010	[18]	3D	chain	4 (4)	✓	DC motor	—	—	—
Vacuubes	Cornell	2011	[98]	3D	lattice	0 (0)	✓	N/A	30x30x30	80g	49

[†] Degrees of freedom (actuated dof in parentheses) [‡] Estimate from figures. — = Information not found in published literature. ✓= yes. ✗= no.

the permutation of male and female connectors around the module's surface only. Each module contains two DC motors, multiple sensors, visual and audio output, and can communicate over Bluetooth and Zigbee. To date (October 31, 2013) the retail price of one Linkbot module is US$190.

2.2 Self-Assembly

Self-assembly is a process in which building blocks arrange themselves into a target shape without external manipulation.The prerequisites for self-assembly of building blocks into target structures are listed in the context of molecular self-assembly by Whitesides et al in [122]: The components must be suitable in scale and material properties for the design of the target structure, and be mobile to change their relative positions. Connections between components must be reversible and attractive and repulsive forces should be in equilibrium. The information controlling the assembly process must be available to the components.

Self-reconfiguring modular robots satisfy these requirements and have the theoretical capability of self-assembly into target structures. A self-assembling a robotic system could be deployed to a remote location without prior knowledge of the task at hand and, upon arrival, self-assemble into a robot morphology determined by environmental variables. Later, the ability to self-reconfigure could be used to autonomously repair the robot or assume a new morphology to achieve a second task.

While self-assembly is a naturally occurring process in chemistry and biological systems that is readily exploited, for example in molecular synthesis, macro-scale self-assembly is a relatively new field, of which Groß and Dorigo give a comprehensive review in [123].

Two approaches to modular self-assembly at the macroscale exist: Deterministic and stochastic self-assembly. In deterministic self-assembly the sequence of motions and assembly operations is planned ahead of time and executed with control over the individual modules throughout the process. Most modular robotic systems capable of self-reconfiguration are capable of deterministic self-assembly when modules are made available in suitable locations. For example, the self-replicating robot by Zykov [20] performs a pre-planned sequence of self-assembly steps during the self-replication process.

Stochastic self-assembly relies on non-deterministic processes to provide actuation the assembly process. Unassembled modules are moved by random actuation, for example on a shaker table or in a mixed fluidic chamber, and the self-assembly process reacts by connecting modules once they appear in suitable locations. Neither the exact sequence nor the duration of the assembly process can be planned in stochastic assembly, but optimal strategies for assembly can be planned and durations estimated [124]. Implementations of two-dimensional stochastic self-assembly are the *Programmable Parts* by Bishop et al. [25, 107], *Stochastic 2D* by White et al. [22], and *XBot* by White et al. [108]. In three dimensions the demonstration of stochastic assembly has been attempted by White et al. [23] and Zykov et al. [125]. No example of multiple actuated robots merging or exchanging modules could be found in literature.

2.3 Self-Refinement

Self-refinement is the improvement of an organism through interactions with the environment that do not require external manipulation such as human intervention. Self-refinement through gathering of information from the environment has been a common

research topic in robotics for decades. For example, in Simultaneous Localization and Mapping (SLAM), robots gather information from the environment with the goal of improving their understanding of the structure of their environment and their position within it. The field of Machine Learning is in part dedicating to devising methods for robots to autonomously self-improve.

Only few instances of physical self-refinement in robotics exist. Bongard et al.'s resilient machines can react to changes in their morphology by adjusting the model their controller is based on accordingly, but cannot change their morphology in order to achieve a target model [126]. Lipson et al. present a system whose mechanical design is initially evolved for target tasks before being automatically manufactured in a 3D-printer, but the resulting robots are final and cannot improve [127]. Zykov et al.'s Molecubes acquire modules from the environment with the purpose of self-replication, making them the only known example of a modular robot acquiring new functionality through collecting building blocks from the environment [20].

CHAPTER 3

SELF-SOLDERING AS A CONNECTION MECHANISM FOR MODULAR ROBOTS[†]

3.1 Introduction

The method applied to forming a connection between the modules of a self-reconfiguring modular robot is widely recognized as key design element of any such system [6,11,76,98,128,129]. The properties of the connectors define the capabilities of the overall robot system: Mechanical properties such as mechanical strength, reversibility, and repeatability define whether a system can be self-reconfiguring; the electrical properties including conductivity and pin count place constraints on the power distribution and communication between modules. This is unsurprising as module-connections play similarly important roles in other modular systems: Molecular bonds play a central role in defining the properties of many engineering materials, and in parallel computing the bandwidth between processing units (modules) can be a limiting factor for performance.

3.2 Requirements

In literature, there is broad agreement on the desirable and required properties of connectors in modular robotic systems [128, 130, 131]:

[†] This chapter includes content previously published in Neubert, J., Rost, A., Lipson, H. (2014) "Self-Soldering Connectors for Modular Robots", IEEE Transactions on Robotics, submitted. This chapter also includes content previously published in Neubert, J., Lipson, H. (2014) "Soldercubes: A Self-soldering Self-reconfiguring Modular Robot System", Autonomous Robots, submitted.

Size With articulated modules housing up to six or eight connectors [11], the size of each connector defines the size of the entire module. Because scalability considerations and actuator torque characteristics generally favor small module sizes, small connectors are desirable.

Mechanical Strength With the primary purpose of a connector being to connect many modules mechanically to form a larger systems, the mechanical strength of the weakest module-to-module connection can define the strength of the overall system. Connectors must function under all loading scenarios encountered during the operation of the modular robot, which might include carrying the load of other modules or externally applied loads.

Information Transmission Capability For a robot to operate as fully self-reconfiguring, information must be shared among the modules without an external intermediary. While wireless information transmission is an option, it is often favorable to transfer information through the connector. This holds true particularly when local communication between neighbors is the primary type of communication in the structure. An in depth discussion of communication in self-reconfiguring modular robots is provided in Chapter 5.

Power Transmission Capability Shared power storage facilities across modules can be beneficial in heterogenous systems where only some modules store energy, or in homogeneous to lessen the momentary demands on batteries during local high current consumption. While wireless energy transfer technology is becoming available in consumer products, for example in inductive cell phone chargers, its low current characteristics and efficiency make it unsuitable for modular robotics. If power transfer between

modules is desired, it must therefore be through the mechanical connector.

Reversibility and Repeatability Self-reconfiguration implies that connections cannot only be formed but also broken. Reversibility is therefore a core requirement for any connection method that is to be used in a self-reconfiguring modular robot. Further, multiple connection-disconnection cycles must be possible to support even basic reconfiguration scenarios. The number of cycles possible before the connection method fails ultimately defines the cost per reconfiguration operation.

Speed of (Dis-)connection While research prototypes normally have little speed performance characteristics, some application do require short connection and disconnection processes. For example, dynamically moving robots such as CKBot [96] can only reconfigure during such a non-static motion if the connections are formed and broken at a time scale equal or smaller than the movements. More generally, faster connector actuation places less restrictions on possible applications and is therefore more desirable.

Tolerance to Error Sources of error in positioning and alignment of modules are manifold: In fully connected modular systems, elasticity, actuator precision, and tolerance of the connection mechanism can affect precision. Where modular robots interact with the environment, external factors further affect relative positioning between modules. It is therefore desirable for the connection mechanism to tolerate such misalignments.

Power Consumption Energy and power availability in each module is limited. When power is distributed, the current and power ratings of connectors themselves are a limiting factor. Without power distribution, the supply characteristics of energy storage and

internal wiring limit the power available to connectors. Typical current rating values are 1 A per pin on a mechanical connector or for one contact of a slip ring[1], and 0.5 A for rechargeable batteries. Naturally, the sum of power requirements of the maximal number of concurrently active connectors may not exceed the supply characteristics.

Genderlessness and Rotation-invariance For self-reconfiguring robots whose modules can take arbitrary poses in space, gendered or rotation-dependent connectors reduce the number of possible configurations. For example, if the connectors of cube shaped modules have a predefined up direction, one can only connect them in six different configurations, while for a rotation invariant connector, there exist 24 possible ways to connect the two modules. Therefore, it is desirable for the connectors to be genderless and rotation invariant. The term *hermaphroditic* is frequently used to describe genderless (or more precisely: two-gendered) connectors, since it was first introduced in the context of modular robotics by Yim in 1994 [38]. Exceptions to this requirement are systems where modules are constraint to one or a few poses, such as the Chobie-II and Metamorphic systems [85], [62].

This chapter describes a connection method previously not used in a modular robotic system, which offers a strong reversible and repeatable mechanical connection that inherently provides electrical connectivity while only consuming power during connection and disconnection.

[1] A slip ring is a connector between two components that rotate relative to each other about the axis of the slip ring.

3.3 Review of Module Connection Methods

Modular robots are machines consisting of a collection of independent self-contained smaller machines of identical or similar type. As opposed to swarm robots, the modules of a modular robot are normally mechanically connected to each other. A subset of modular robots has the property of self-reconfigurability, meaning that through appropriate actuation the modular robot can change the arrangement of its own modules without external manipulation. The latter implies that mechanical connections between modules must be able to form and break in a controlled manner.

The types of connectors presented in literature to date can be grouped broadly into mechanical, magnetic, and non-articulated connectors. In order to provide context for the development of a new modular robot connector in this chapter, the existing solutions to the connection problem are discussed below and Table 3.1 lists properties of the connectors for several self-reconfiguring modular robots.

3.3.1 Mechanical Connections

In a survey of connection methods for self-reconfiguring modular robots, mechanical connectors are by far the most common.

Grippers are systems where an active end effecter-like mechanism grips a counterpart or passive structure on a neighboring module to connect. Self-reconfiguring robot systems that implement gripper mechanisms include *ATRON*, *Molecule*, *3D Unit*, and *CoSMO* [66,79,86,93]. Gripper connectors are mechanically strong and reliable enough to be used in an assembly of 100 *ATRON* modules. However, they add significant mechanical complexity to the module. Either each actuated connector requires an inde-

pendent drive mechanism, or a complex clutch mechanism is required to actuate the connector independently [66]. As a result, modules with gripper connectors often have few connectors or are large.

Latched connectors are mechanical connectors where connections are formed passively but disconnection requires actuation. The *Crystalline*, *I-Cubes*, *Chobie-II*, *MTRAN-III* and *Micro Unit* all slide a grooved pin into a spring loaded lock to connect, and disconnect by releasing the lock through either SMA or DC motor actuation [85, 109, 132–134].

A variation on the latched connector type are those where multiple pins are unlocked, and in some cases also locked, in parallel by rotating two connector plates relative to each other. This type of connector can be found in the *CONRO*, *ModRED*, and *Roombots* systems [16, 89, 135, 136]. ModLock is a connection for *CKBot* based on the same principle but manually operated and has been shown to support loads of up to 2.2 kN [137].

Table 3.1: Review of Connection Methods for Modular Robots

Name	Year	Dim	dof	Connection Type	Count	Strength	g†	P	S	Size (mm)	Weight	Ref.
CEBOT	1988	2D	N/A	SMA actuated latch	2 (1)	—	✗	✗	✓	180x90x50	1.3kg	[41]
Metamorphic	1993	2D	3 (3)	sliders	6 (3)	—	✗	✗	✗	—	—	[62]
Polypod	1993	3D	2 (2)	SMA actuated latch	2 (2)	—	✗	✓	✓	—	—	[60]
Fracta	1994	2D	1 (1)	electro-magnet	3	—	✗	✗	✗	D:125 H:160	1.2kg	[2]
Metamorphic2	1996	2D	2 (2)	sliders	6 (3)	—	✗	✗	✗	—	—	[62]
CONRO	1998	3D	2 (2)	SMA actuated latch	3 (1)	—	✗	—	✓	L:108	115g	[16]
Micro-Unit v1	1998	2D	2 (2)	SMA actuated latch	4 (2)	—	✗	✗	✓	50x50x50	50g	[133]
Molecule	1998	3D	4 (4)	gripper	10 (10)	—	✗	—	—	D:102	3.2kg	[76]
3D-Unit	1998	3D	6 (6)	gripper	6 (6)	—	✓	✓	✓	L:265	7kg	[66]
Vertical	1998	2D	2 (2)	permanent magnet	5 (1)	0.3N	✗	—	—	90x90x90	—	[101]
I-Cube	1999	3D	3 (3)	key and lock	2 (2)	—	✗	✗	✓	85x37x18	205g	[103]
Crystalline	1999	2D	1 (1)	key and lock	4 (2)	—	✗	✗	✗	51x51x178	340g	[102]
Micro-Unit v2	1999	2D	2 (2)	SMA actuated latch	4 (2)	—	✗	✗	✓	20x20x30	15g	[138]
PolyBot	2000	3D	1 (1)	SMA actuated latch	2 (2)	—	✓	—	✓	50x50x50	200g	[3]
MTRAN	2000	3D	2 (2)	SMA act. p. m.	6 (3)	25N	✗	✓	✓	66x132x66	440g	[139]
Telecubes	2002	3D	6 (6)	SMA act. p. m.	6 (6)	—	✓	✓	✓	60x60x60	—	[104]
Chobie-II	2003	2D	2 (2)	sliders	4 (2)	—	✗	✗	✗	80x80x75	500g	[26]
Gear-Unit	2003	2D	1 (1)	permanent magnet	6 (0)	—	✗	✗	✗	D:55	—	[105]
MTRAN-II	2003	3D	2 (2)	SMA act. p. m.	6 (3)	—	✗	✓	✓	60x120x60	400g	[82]
Stochastic 2D	2004	2D	0 (0)	DC motor act. p. m.	3 (3)	—	✓	✗	✗	60x60	100g	[22]
Catoms	2005	2D	0 (0)	electromagnets	24	—	✓	✓	✗	D:44 H:40†	105g	[106]
Stochastic 3D	2005	3D	0 (0)	perm. & electromag.	6 (6)	—	✓	✓	✓	100x100x100	—	[23]
Molecubes I	2005	3D	1 (1)	electromagnets	2 (2)	—	✓	✓	✓	10x10x10cm	625g	[21]
ATRON	2005	3D	1 (1)	gripper	8 (4)	200N	✗	✓	✗	D:11	850g	[87]
Prog. Parts	2005	2D	0 (0)	DC motor act. p. m.	3 (3)	—	✓	✗	✗	L:120 H:42	—	[107]
XBot	2007	2D	0 (0)	SMA act. p. m.	4 (4)	—	✓	✗	✗	—	—	[108]

Table continues on next page

Table 3.1 continued

Name	Year	Dim	dof	Connection Type	Count	Strength	g†	P	S	Size (mm)	Weight	Ref.
MTRAN-III	2008	3D	2 (2)	DC motor act. hook	6 (6)	—	✗	✓	✓	65x65x130	420g	[109]
Roombots	2010	3D	3 (3)	DC motor act. latch	10 (2)	—	✓	✗	✗	220x110x110	1.4kg	[131]
ModRED	2010	3D	4 (4)	solenoid act. latch	2 (2)	—	✓	✗	✗	368x114x1190	3.17kg	[92]
M3 Express	2012	2D	3 (3)	DC motor act. p. m.	3 (3)	11N	✓	✗	✗	—	878g	[111]
CoSMO	2013	3D	1 (1)	gripper	4 (4)	4kN	✓	✓	✓	105x105x105	1250g	[140]
M-Blocks	2013	3D	1 (1)	permanent magnet	6 (0)	—	✓	✗	✗	—	—	[141]

† **g** = genderless? **P** = Transmits power? **S** =Transmits signal? ✓= yes. ✗= no. — = Information not found in published literature.

3.3.2 Magnetic Connections

Using magnetic force is a second approach that is frequently used for forming connections between robot modules. Magnetic module connectors can be broadly categorized by their use of electromagnets, static permanent magnets, and actuated permanent magnets.

Static permanent magnets are only usable for reconfigurable modular robots, but not self-reconfiguring systems. Because static permanent magnets' attraction and repulsion force cannot be controlled by the modules, disassembly of a connection without external manipulation is not possible. For example, *Cubelets* form connections through permanent magnets embedded in the module surfaces and rely on human manipulation for disconnecting modules.

Active permanent magnet connectors are connectors in which permanent magnets are connected to mechanical actuators. A mechanism where shape memory alloy (SMA) wire is used to retract permanent magnets from the connected position is used in the first two generations of the *MTRAN* system, *XBot* and the *Telecubes* system [55,82,104,108]. In the *MTRAN* system, for example, the magnets provide sufficient force to support a force of 3.6 kg that, internal to the module, is nearly balanced by a spring. When the SMA wire is heated to exert a small force on the magnet which, in series with the spring force, pulls the magnet away from the connector surface, thereby reducing the connector force resulting in disconnection. *M3Express* and *Programmable Parts* both implement a similar connector, but actuated by small DC motors [107, 111].

An alternative way to use mechanical actuation to disconnect permanent magnet connectors is to change the relative polarity of adjacent magnets. A commercial implementation of the actuated permanent magnet concept is available under the brand

name Magswitch® from Magswitch Technology, Inc, and has been used in the *Miche* system [24].

Electro-permanent magnets provide an interesting alternative that overcomes those problems, which has recently been applied to modular robotics. In an electro-permanent magnet an electromagnet surrounds two permanent magnets of different magnetic coercivity. A short current pulse through the electromagnet is required to change the polarity of the low coercivity permanent magnet. Flux circulates within the electro-permanent magnet device when the polarity of the two permanent magnets is reverse, but when the polarities are switched to be equal, the device acts as a permanent magnet. This means that an electro-permanent magnet has low power requirements similar to permanent magnets but the switchability of an electromagnet [142]. The *Pebbles* self-disassembling system makes use of electro-permanent magnets [143, 144].

Compared to mechanical methods connection methods, using magnetic force for the connection method has the benefit that magnetic force acts over a distance, making the magnet connecter useful during the alignment and docking process without specifically designing for it. Disadvantages of magnetic connectors are the lower connection force and, for electromagnets, the high current consumption, which is traded off in active permanent magnet mechanisms with mechanical complexity.

3.3.3 Connections with Binder Materials

More recently, a number of connection methods that use a phase changing binder material between connecting modules have been reported. Miyashita et al. use Peltier elements to freeze water between adjacent connector surfaces [145]. This connector requires constant power input to sustain cooling to retain the connection. Wang et al.

use hot melt adhesives that require power only to be heated in order to form or break connections [146].

The soldering connector described in this Chapter uses a binder material to form connections, too. It extends the methods using adhesives and water by providing superior mechanical strength and electrical conductivity for signal and power transfer.

3.3.4 Other Connections

To give a complete picture of connection methods in modular robotics, two more connection types should be mentioned. *Vacuubes* are manually reconfigurable modules that, when assembled into a passive structure, are connected to a vacuum pump, resulting in the differential pressure to atmosphere holding the structure together [98].

For the *YaMoR* system, Velcro has been used as a static connection method, which has the same benefits and drawbacks as static permanent magnets, namely that disconnection without external manipulation is not possible [147].

3.4 Component Selection

The thermally actuated self-soldering connecter has two central components: The low melting point alloy, and the heating element. In support of their function several other components are further required: A carrier to hold or contain the low melting point alloy and a switch to toggle heating. Additionally, during the assembly of the self-soldering connector, the application of low melting point alloy to most carriers will require a flux. The following sections discuss the selection of each of those components.

Table 3.2: Melting Points for a Selection of Low Melting Point Alloys

Composition (by weight)/Trade Name	**Melting Point** (°C)
Mercury	−39
68 % Ga, 22 % In, 10 % Sn ‡ (GalInStan)	−19
GaInSn alloys †	< 30
Ga	30
BiPbInSnCd alloys	> 43
51 % In, 32.5 % Bi, 16.5 % Sn‡ (Field's Alloy)	62
66 % In, 34 % Bi	72
69 % Bi, 26 % In, 17 % Tn	72
63 % Sn, 37 % Pb ‡ (Electronics Solder)	183

† Exhibits supercooling. ‡ Eutectic.

3.4.1 Solder

The key component of a soldered connection is the solder. For our application we are looking for a conductive material that is solid at the normal operating temperature range, but can reach its melting point with little energy input as possible. Table 3.2 lists commonly available low melting point alloys and their melting points.

The power required to heat a substance with specific heat c and mass m over time t by ΔT °C is

$$P = \frac{cm}{t}\Delta T \tag{3.1}$$

If the substance is exposed to atmosphere like it is in the connector application proposed here, a convective heat loss of

$$\frac{dQ}{dt} = -hA\Delta T(t) \tag{3.2}$$

occurs at all times. It follows that in order to melt the substance, the power input

must at least compensate the convective heat loss at the melting temperature. Further, the higher the power input, the shorter the time to heat the substance to its melting point, which equates to a faster connector actuation for our application. For a specified connector actuation time, and assuming that the least amount of solder possible is used, the only means of reducing the power requirement of the connector is to use a solder with low melting temperature.

The convective heat loss equation offers potential opportunities and challenges: One could select the atmosphere in which the soldering connector operates to minimize heat transfer and thereby reduce the power requirements of the connector. We chose to pursue the opposite path and make the connector suitable for operating submerged in liquids, as applicable to fluidic stochastic assembly robots such as the ones by White et al. [23] and Zykov et al. [125] and the system described in Chapter 6.

The melting point of the 63 % Sn, 37 % Pb solders commonly used for electronics assembly is above 180 °C, and above 200 °C for lead-free solder. Even lowest power soldering irons used for these solders draw 30 W of power. In a robot module this power would need to be supplied or transferred through individual modules. However, 30 W far exceeds the supply capability of every battery pack as well as the specifications of many components available at the scale of a self-reconfiguring modular robot module.

When seeking for candidate materials with lower melting points two additional design requirements apply. First, a non-toxic material is desirable. Alloys containing Mercury or Gallium do not meet this requirement due to their known health effects. Second, it is desirable for solders to have a eutectic alloy composition. This means that the alloy melts at a single temperature, whereas non-eutectic alloys exist as a paste of molten and solid components throughout a melting temperature range.

Of the items listed in Table 3.2, 51 % In, 32.5 % Bi, 16.5 % Sn alloy meets both requirements and has a melting temperature of 62 °C. The metal is also know under the name Field's Alloy[2]. Field's Alloy is chosen as the solder for use in the self-soldering connector.

3.4.2 Solder Carrier

The Field's Alloy must now be applied to an exterior surface of the robot module where it can be brought into contact with a matching surface of a neighboring module to form a soldered connection. In principle, any surface that brings two masses of Field's Alloy in contact works, but printed circuit boards (PCB) are likely to be most suitable. PCBs are designed to carry solder, there exist well established manufacturing and design methods that makes them easy to customize, and PCBs have a smallest space requirement compared to other not flat design options. In our experiments, we use a 0.032in thick FR4 printed circuit board with a series of exposed tinned copper pads on one side, to which Field's Alloy will be applied.

The specific design of the Field's Alloy carrying PCB depends on the application and some design considerations are discussed below in Section 3.5.2.

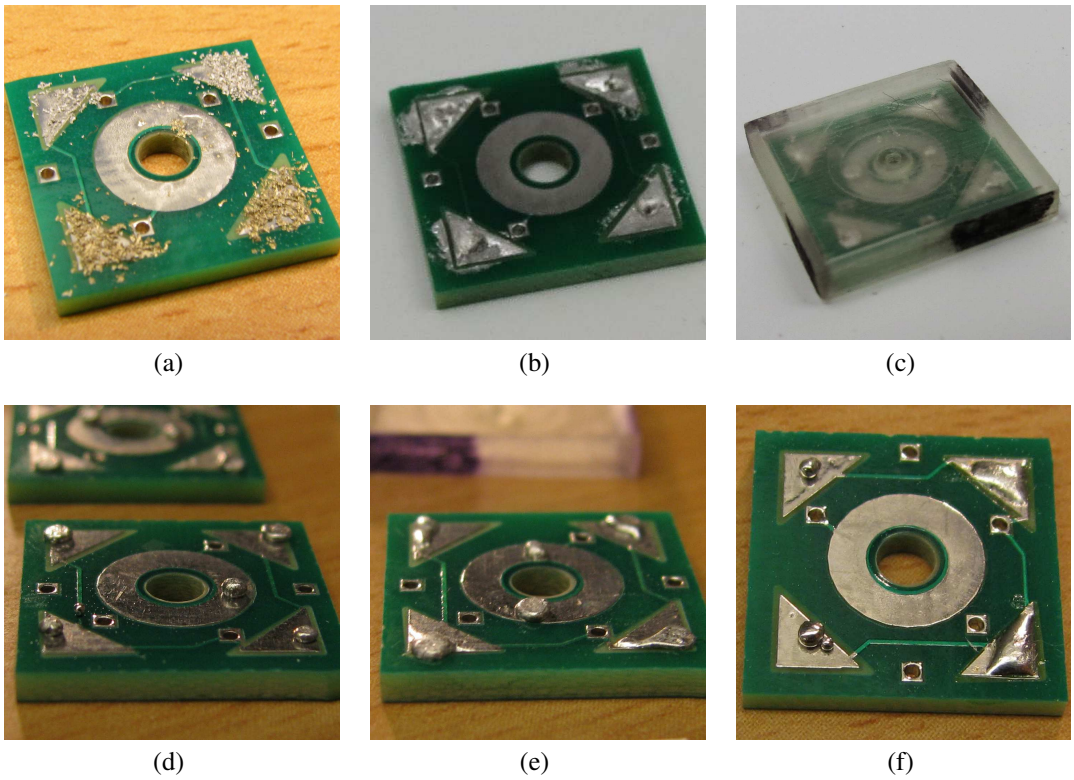
[2]The name "Field's Alloy" was coined by accident: The inventor and author Simon Quellen Field was seeking a non-hazardous low melting point alloy for instructional tutorials on his website www.scitoys.com. When ordering a candidate material, the foundry needed a name for their paperwork and simply used the customer name. Mr Field decided to use the new name on his website and soon thereafter it appeared in scientific publications as "Field's Alloy" and "Field's Metal" [148].

3.4.3 Flux

An undesirable property of Field's Alloy is its high surface tension and low reactivity with other metals. This makes the application of Field's Alloy to the carrier PCB difficult: Figure 3.1(a) shows the results of attempting to apply Field's Alloy to tinned copper pads creating an alloy powder by abrasion and applying it in an oven. Figure 3.1(b) shows the result of using GalInStan (68 % Ga, 22 % In, 10 % Sn) as wetting agent; similar results were observed with pure Gallium. Figures 3.1(c) to 3.1(e) show the results of attempts to use 3D-printed molds to bring Field's Alloy in contact with tinned pads of PCBs. Finally, Figure 3.1(f) shows first promising results achieved through use of rosin flux: The flux-treated pads are coated more evenly with alloy, a prerequisite for reliable module connections.

Fluxes are commonly used in electronics assembly and other metal joining applications to pretreat metallic surfaces before applying solder. The goal is to remove impurities from the soldering pads and to prevent oxidation, allowing the solder to adhere to the pad. Fluxes vary by their composition, activation temperature, and residue type. Selection of the correct flux for a combination of surface and solder composition is usually based on empirical data and experience. Flux application consists of four steps:

1. Application of the flux to the solder surface (the PCB's copper pads) by dipping, spraying, brushing, or otherwise.
2. Activation of the flux by heating to its activation temperature.
3. Removal of excess flux from the solder surface to avoid "sputtering" (splashing of flux during solder application)
4. Solder application



(a) (b) (c)

(d) (e) (f)

Figure 3.1: Photographs of failed attempts at applying Field's Alloy to PCBs. (a) Application by baking at 120 °C (top) and 240 °C (below) as powder. (b) Using GalInStan as a wetting agent results in somewhat improved coverage but GalInStan also adheres to the soldermask. (c) Using a mold filled with Field's Alloy pellets results in no improved coverage for a mold with 0.3 mm cavity height (d) and slightly improved coverage for a mold with 0.1 mm cavity height. (f) The use of rosin flux (right) provides some improvement over no flux usage (left) but does still not yield complete coverage of the exposed pads.

5. Cleaning of the flux residue. Depending on the formulation of the flux, this is usually either achieved through washing with warm water or an alcohol acetate solution.

To find a suitable flux for applying Field's Alloy to tinned PCB, a candidate selection of six fluxes listed in Table 3.3 was tested. In an exploratory series of experiments several process parameters were varied for each flux. In these experiments, flux was

Table 3.3: Selection of Candidate Fluxes

Manufacturer	**Product Name**	**Activation Temp.** (°C)	**Cleaning**
Kapp	Golden	175 – 280	warm water
MG Chemicals	Rosin Flux	90 – 205	alcohol acetate mix
Indium Corp	Indalloy Flux #2	100 – 371	warm water
Superflux	No 30	95 – 315	warm water
Superflux	No 75	95 – 345	warm water
Superflux	No 79	95 – 315	warm water

applied to the PCB using a cotton swab and then let stand for a specified application time at room temperature. Subsequently the PCB was either cleaned or not (depending on the experiment), and dipped upside-down into a bath of molten solder. Variables explored in the experiment were:

- Solder temperature: The temperature at which the vat of molten solder is kept.
- Application time: The time between flux application and solder application, either $10s$ or $30s$.
- Pre-cleaning: Whether excess flux was removed from the PCB by wiping before the solder application step.
- Flux selection.

In addition, two control experiments were carried out: No flux application, and flux application to the surface of the heated solder vat instead of the carrier PCB. The latter was considered because visible buildup and oxidation occurred on the surface of the molten solder vat which would disappear when fluxed, but *not* pre-cleaned PCBs were dipped into the solder.

For each parameter set, five PCBs were processed with the goal of applying solder to all circuit board pads. To analyze the results quantitatively, the coverage of each PCB

Table 3.4: PCB Pad Coverage Classification

Value	Description
0	No wetting, bare pad
1	Up to 1/4 of pad covered in solder
2	Up to 1/2 of pad covered in solder
3	Up to 3/4 of pad covered in solder
4	Complete solder coverage with contact angle of <45°
5	Complete solder coverage with contact angle of >45° suggesting excessive solder application

pad was classified on a scale from 0 to 5 as explained in Table 3.4.

Results from twenty seven experiments with five PCBs were analyzed in Figure 3.2. Three experiments were excluded from analysis, because pre-cleaning was performed in an uncontrolled fashion, rendering the results non-interpretable. Further, the top of Figure 3.2 displays examples of incomplete or incorrect solder application. It is apparent that every flux, given appropriate process parameters, can be used to improve the solder coverage on the circuit board when compared to no flux use at all. *Superflux 75* was identified as showing the most promising results and a broader spectrum of solder temperatures was investigated. It provides reliable complete coverage of all PCB pads with some process parameters leading to the application of too much solder per pad, represented by average coverage numbers larger than four in Figure 3.2. Appendix C contains top and side views of all 150 PCBs tested including experiments 7 to 9 not included in the analysis.

Besides incomplete solder coverage, several other error modes occur: Some parameter sets result in consistent or inconsistent buildup of large amounts of solder on individual pads. This is undesirable because it leads to short circuit connections between neighboring pads, either during the cooling period immediately after solder application or later during operation of the connector. Some parameter sets result in solder attraction to non-metallic surface sections of the PCB instead of the exposed copper sections.

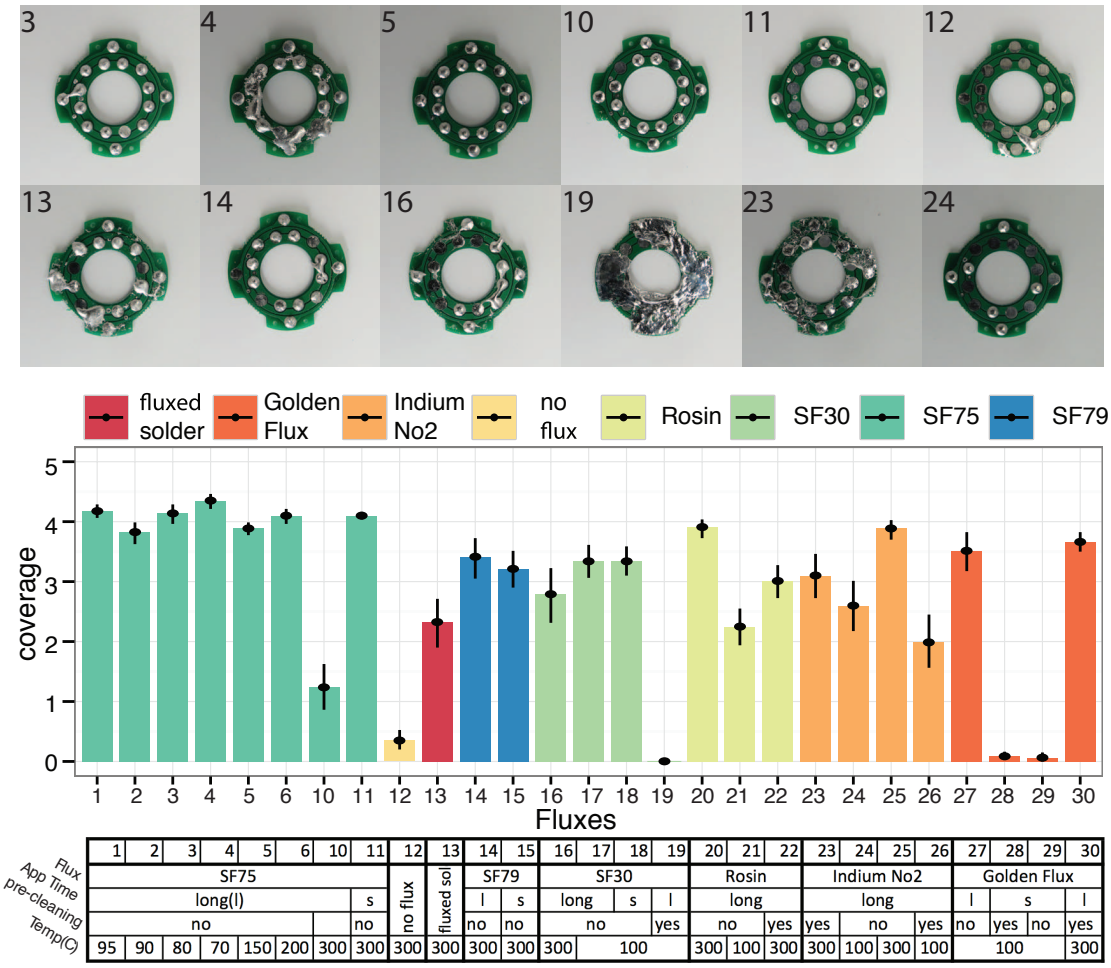


Figure 3.2: Analysis of 27 experiments of the application of Field's Alloy with various flux and application processes.

The parameters *pre-cleaning* and *flux application time* are considered separately in Figure 3.3(a), each in comparison to no flux application and flux application to the solder vat as reference values. On average, as well as for Superflux 75 only, omitting pre-cleaning and waiting for the longer period of application time results in improved solder pickup on the PCB pads.

Figure 3.3(d) aggregates the solder pad coverage by pad location over all experiments, with all flux types and process parameters. This is to determine whether the position of the pad affects the wetting of Field's Alloy. No significant variation of solder

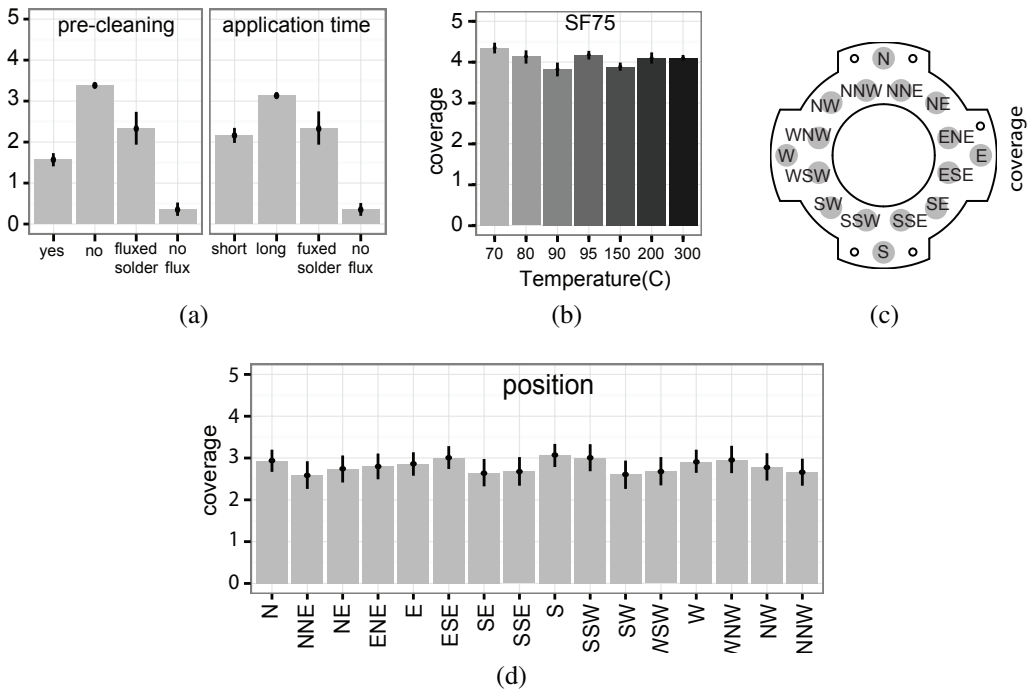


Figure 3.3: Analysis of solder coverage on PCB pads for selection of fluxes and process parameters. (a) Experiments aggregated by pre-cleaning parameter (left) and application time parameter (right). (b) Experiments with Superflux 75 analyzed by solder temperature. (c) Legend of solder pad location naming used. (d) Solder coverage analyzed by circuit board pad location with data aggregated from all experiments. No significant variation of coverage with position.

coverage is present suggesting that the position on the PCB does not affect how well solder is picked up when dipping the PCB into the molten solder. However, qualitative observations suggest that the amount of solder applied to the outer pads (N, E, S, W in Figure 3.3(c)) is smaller than that on the inner pads.

The solder temperature does not have a significant effect on the solder coverage as indicated by the results in Figure 3.3(b), which aggregates all experiments that use Superflux 75. This suggests that the flux improves the solder wetting compared to no flux and all other tested fluxes, even though it never reaches its activation temperature, neither before solder application nor during solder application. This result would warrant

further investigation, but here it is simply noted that there is no disadvantage of heating the solder to higher temperatures. Superflux 75 with a solder temperature of 150 °C to 200 °C presents the final flux selection. Its application details based on the date presented here is discussed in the following Section 3.5.

3.4.4 Resistive Heater

Common choices for heat generating components are Peltier elements and resistive heaters. Peltier elements have property of reversing heat flow and acting as cooler in reverse polarity which could be useful to accelerate solidification in the soldering connector. However, Peltier elements are far more expensive and larger than resistors, which heat under both polarities. Further, Peltier elements are not readily available as surface mount components impeding manufacturability. Therefore, resistors are chosen as the source of heat for melting Field's Alloy in the self-soldering connector.

All electrical power absorbed by a resistor is converted into heat which must be released to the environment. Normally it is a design goal in circuit design to minimize the heat generated, and spread it efficiently in order to keep the electrical circuit from overheating. For the application of melting a substance on the same PCB, the objective is instead to generate as much localized heat as possible with a resistor.

Ohm's law $V = RI$, and $P = VI$ suggest that for any given supply voltage one wants to minimize the resistance of the heating element to convert the maximal amount of power into heat. Assuming, as a thought experiment, unlimited current supply, this strategy is constrained by the ability of the resistor to reject heat into the environment. If the resistor absorbs more power than it can reject as heat into the environment, the internal temperature rises and the resistor will fail. The thermal properties of resis-

tors are commonly described by the parameters of *maximal power rating*, and *maximal operating temperature*. The maximal operating temperature specifies at which temperature the measured resistance will fall outside the specified range, either permanently or reversibly. For the purpose of the soldering connector, it is only important that the maximal working temperature is higher than the melting temperature of Field's Alloy, which is the case for all commercially available resistors. The maximal power rating is more informative: Because all absorbed energy must be balanced by heat rejection into the environment, the maximal power rating indicates the resistor's ability to transfer heat towards the outside of the resistor package.

The number of surface mount resistor packages on the market is vast. Specialty resistors with extremely high power ratings exist for some applications: For example Bourns, Inc, offers resistor for high power radio-frequency applications in common surface mount component package sizes such as a 1206 with 20 W power rating while typical power ratings for similar sized resistors are below 1 W.

In addition to a significant price premium these parts have non-standard solder footprints or require additional metallic clamps to improve heat transfer. Both price and large footprint mounting requirements are the reason specialty high power resistor packages are not suitable choices for the application of a scalable robot module design. The choice of standard solder footprint resistor packages has the benefit of making most efficient use of PCB area and allows for simple automatic assembly.

First iterations of the soldering connector used the highest power rated 2512 package resistor with standard solder footprint available (part no CRM2512-FX-10 by Bourns, Inc, rated at 2 W). Subsequent tests confirmed that even cheaper resistors with common power ratings of 1 W and 0.5 W remain functional after being repeatedly subjected to small multiples of their power rating.

Following the investigations described above, the final selection process for the heating element of the soldering connector consists of selecting a standard footprint 1 W rated resistor at the desired resistance and size. The choice of resistance and package size depends on size and other parameters of the specific implementation and is not discussed here. Section 3.5.2 discusses detailed component selection for a sample design of the soldering connector that is suitable for modules with connector surface areas of approximately 1 in^2.

3.5 System Design

With the selection of components now in place, the following sections serve as a recipe for assembling a PCB to serve as the connector between robot modules. In addition to discussing general design considerations, a reference design for a soldering connector with a surface area of approximately 1 1/4 in^2 (820 mm^2) is described.

3.5.1 Heat transfer considerations

Section 3.4.4 discusses the selection of standard footprint thin film surface mount resistors as the heating elements for the soldering connector. These resistors are to be mounted on the same PCB chosen as the carrier for the low melting point alloy, see Section 3.4.2. Because no part of the connector board can extend beyond the outer surface of the PCB where the low melting point solder is applied, the resistors must necessarily be mounted on the opposite side of the PCB resulting in an arrangement as shown schematically in Figure 3.4.

In thin film resistors the resistive element is a thin layer of conductive material, the

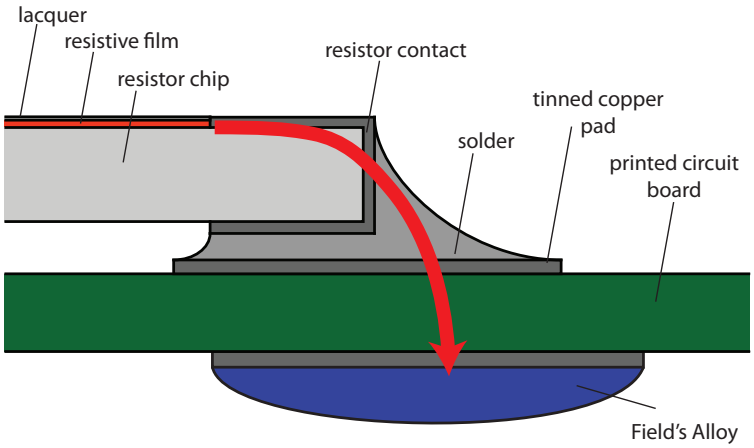


Figure 3.4: Schematic view of heat transfer through a PCB with a surface mount resistor mounted on one side and a Field's Alloy covered solder pad on the other side.

"film", on top of a substrate referred to as "chip", which accounts for most of the thickness of the resistor package. The most common termination for surface mount resistors are metallic caps that wrap around the short sides of the package and are electrically and thermally connected to the film. During PCB assembly a solder joint is formed between each resistor terminal and a tinned copper pad on the PCB, forming an electrical and thermal connection between the resistor and the circuit board. The total thermal resistance of the conductive heat transfer path from the heated resistive element to the Field's Alloy can therefore be written as a series of thermal resistances[3]:

$$R_{total} = R_{res} + R_{sol} + 2R_{cop} + R_{sub} \tag{3.3}$$

where the individual resistances are

- R_{res} the thermal resistance of the resistor itself including film and chip.
- R_{sol} the thermal resistance of the solder joint between resistor and PCB

[3]This representation is adapted from the more detailed characterizations of heat transfer in surface mount components by Mauney [149] and Visahy Intertechnology, Inc [150].

- R_{cop} the thermal resistance of the copper layer, both top and bottom of the PCB
- R_{sub} the thermal resistance of the PCB base material, using as identifier the insulation grade *FR4*, which is frequently used to refer to the PCB substrate material.

Additional heat transfer through radiation and convection is negligibly small in this application.

In order to melt the Field's Alloy quickly, as well as to prevent thermal damage to the resistor, the total thermal resistance R_{total} between the resistor film and the Field's Alloy must be minimized.

R_{res} is minimized by choosing a resistor with large power rating as discussed in Section 3.4.4.

R_{sol} could be reduced further by choosing a higher conductivity solders such as 96.5 % Sn/3.5 % Ag ($k_{th} = 78\,\mathrm{W\,m^{-1}\,K^{-1}}$), but this strategy was not pursued for the reference implementation because of the minimal potential benefit paired with the very limited availability for this and other unconventional solders at low volume PCB assembly suppliers. The thermal conductivity k_{th} of leaded 63 % Sn/37 % Pb leaded solder is is quoted as $51\,\mathrm{W\,m^{-1}\,K^{-1}}$ [151]. Compared to k_{th} for the other materials in the PCB assembly, this value is very high, and the low ratio of solder joint thickness to surface area contributes further to making R_{sol} a negligible component of R_{total}.

After choice of resistor and solder have been addressed, the design features of the fabricated PCB before assembly, the copper layer and the substrate, are left as parameters for minimizing the thermal resistance of the assembly. The PCB substrate material, epoxy-infused fiberglass, has the highest thermal resistance of all components in the PCB assembly with a thermal conductivity of $0.16\,\mathrm{W\,m^{-1}\,K^{-1}}$ [151]. This makes improving the conduction through the PCB substrate the most effective way to reduce R_{total}.

A simple way to reduce the thermal resistance of the substrate is to reduce its thickness. For the soldering connector we chose 0.8 mm thickness substrate, a tradeoff between reducing thermal resistance and ensuring sufficient mechanical strength. The ideal PCB thickness for other implementations of the soldering connector might differ depending on other application requirements: PCBs can be manufactured with substrates as thin as 0.5 mil (12.7 μm), with values below 100 μm considered "flexible PCB".

Plated holes, or "vias", are a strategy to improve heat transfer between copper layers of a circuit board by creating a low thermal resistance metallic connection. However, because such thermal vias also form an electrical connection, and because the resistive heater terminals are not connected to the soldering connector connection pads in the circuit, a direct metallic connection between most heating resistor terminals and the Field's Alloy is not possible[4]. While this does not rule out the use of thermal vias between the two sides of the PCB, their utility would be limited because gap of the minimal trace separation width specified by the PCB manufacturer still needs to be left to electrically insulate the copper traces. Further, if the vias are not covered by soldermask, either by design or through wear and tear on the solder mask, any additional via on the Field's Alloy side of the PCB increases the chance for unintentional short circuits when Field's Alloy leaks from the pads it has been applied to.

An alternative to thermal vias connected to the copper on the heater side of the PCB are thermal vias embedded in the Field's Alloy covered pads and terminating underneath the resistors but not electrically connected to the resistor terminals. To facilitate heat transfer on the heater side of the PCB one could consider the application of thermal paste underneath the resistor. Figure 3.5 shows that this design results in the Field's

[4]There exists one exception: If the supply voltage to the heaters is connected to the power and ground pins of the connector, two of the resistor terminals are at the same voltage level as 12 out of 16 connector pads in the reference implementation and could be connected to those pads, space and other layout constraints permitting.

Figure 3.5: Photograph of soldering connector PCB prototype with thermal vias inside the Field's Alloy covered contact pads. After applying Field's Metal through various application methods, the embedded vias result in incomplete Field's Alloy coverage.

Table 3.5: Guide for Choosing Printed Circuit Board Design Parameters

Design Parameter	**Selection Considerations**	**Reference Implementation**
Substrate Thickness	As thin as permitted by stiffness requirements	0.8 mm
Copper Weight	As high as possible	2.0 oz in^{-1}
Copper Coverage	As high as possible	approx. 75 %
Thermal vias	Where possible given circuit design	none
Type of vias	Outside Field's Alloy pads, soldermask covered	N/A
Solder Selection	High thermal conductivity is preferred	63 % Sn, 37 % Pb

Alloy not covering the thermal via in many cases, and experiments showed that while forming a connection Field's Alloy can leak through the via and cause unintentional electrical connections and short circuits on the heater side of the PCB.

With the only possible optimization of the PCB substrate design being the choice of a relatively thin 0.8 mm substrate, the final remaining means of reducing the thermal resistance R_{total} of the PCB assembly is increasing the surface area of PCB substrate and decrease the thermal resistance R_{cop} of the copper layers on either side of the PCB. Both is achieved by placing the maximal volume of copper on both top and bottom side of the PCB, while staying within the necessary constraint of electrical validity of the resulting circuit. Figures 3.7(a) and 3.7(b) show the copper layer of the reference implementation PCB with approximately three quarter of the surface area covered with copper. The thermal resistance of the copper layer itself, R_{cop}, is further reduced by choosing a high

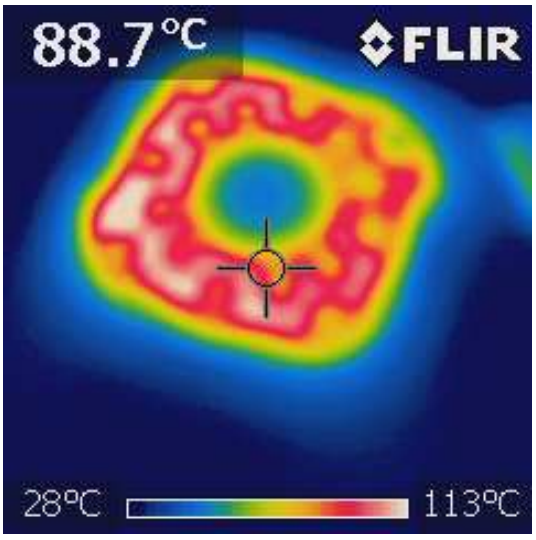


Figure 3.6: Thermal image of PCB during heating. Image recorded with a FLIR f7 thermal camera.

copper weight of 2.0 oz in^{-1}, equating to approximately 70 µm thickness. Depending on the constraints of the PCB manufacturer chosen, copper weights up to 5.0 oz in^{-1} may be available, though care needs to be taken that R_{cop} is not reduced as far as impairing the ability of the PCB assembly process to heat the pads on the heater side of the PCB while placing the resistive heaters.

Table 3.5 summarizes which PCB design parameters can be optimized for heat transfer between heaters and Field's Alloy covered connector pads. Figure 3.6 looks at the Field's Alloy side of one reference design PCB after heating for 15 s continuously at 12 V in air, resulting in a surface temperature of approximately 70 °C to 110 °C on the PCB surface including the Field's Alloy covered pads.

3.5.2 Other PCB Layout Considerations

In addition to the aforementioned heat transfer considerations, several other design constraints may need to be considered in the design of the soldering connector's PCB:

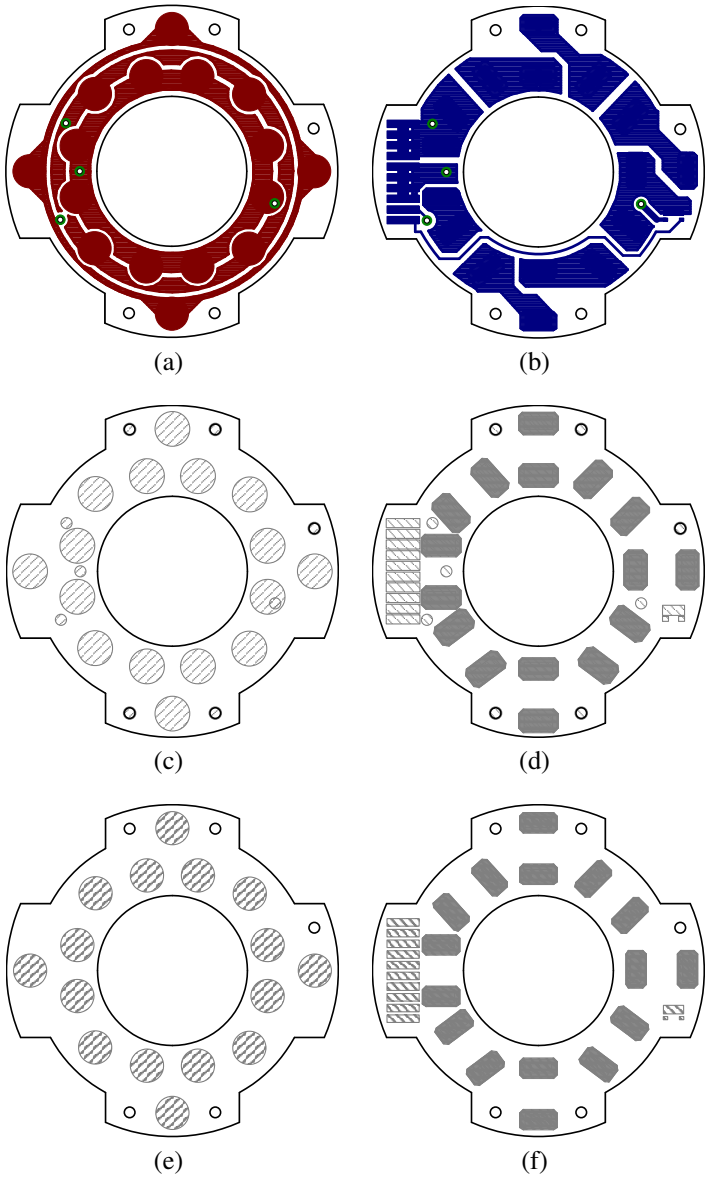


Figure 3.7: PCB layout drawings for an implementation of a self-soldering connector with approximately $11/4\,\text{in}^2$ ($820\,\text{mm}^2$) connector surface area. (a), (b): Copper layer, red top, blue bottom, green plated holes. (c), (d): Soldermask stop layer, hatched and shaded regions define areas of exposed copper layer. (e): Stencil layer top, regions where fields metal application is intended. (f): Stencil layer bottom, regions where solder paste is deposited during assembly of resistive heater and other circuit components. (a), (c), (e) represent the Field's Alloy side of the PCB, (b), (d), (f) the heater side. All figures show the PCB outline and five non-plated drill holes.

- The mechanical strength required by the application and the assembly process limits how thin the PCB can be.
- Solder selection and copper weight are limited by the capabilities of the PCB assembly process chosen, particularly by the maximal temperature and power specification of the soldering method.
- Selection and limits of PCB thickness and availability of soldermask-covered thermal vias are limited by the capability of the PCB manufacturing process, with higher price or larger quantity making more flexible processes available.
- Placement and size of holes for fasteners or an alternative fastening method to hold the connector PCB in place.
- Selection and placement of a heater switch, for example a MOSFET device, to control heating of the Field's Alloy. To reduce the number of current carrying connections to the soldering connector, it makes sense to use the same power and ground lines that connect to the connector pads for transferring power to neighboring modules, to power the connector's heater. This necessitates placing the switch to control the heating on the connector PCB. When doing so the temperature specifications of the device and the solder used to connect need to be taken into consideration in order to prevent damage or disconnection of the device during heating.
- The outer shape of the PCB is driven almost entirely by the application and other factors of the module design where multiple features might compete for surface area on the module's outer surface. When connectors are rotation-invariant it can be useful to introduce an asymmetry into the outward facing design to aid with testing and debugging where one might not have access to the robot module's interior. The reference design introduces such an asymmetry through the fastener hole placement, though simple silkscreen markings would serve the same purpose.

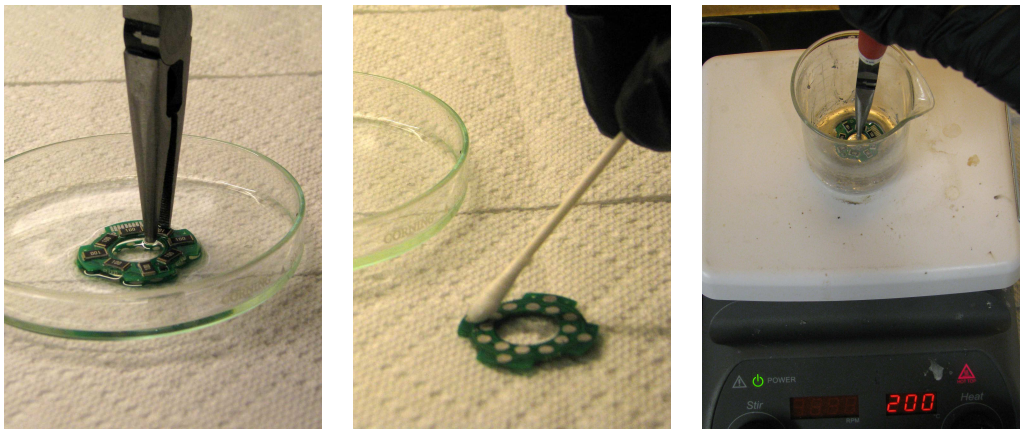
- The same design rules that apply to all PCB manufacturing naturally also apply to the soldering connector PCBs: For example, a significant price saving in the manufacturing and assembly step could be achieved by panelizing the PCBs in batches of nine. Generally applicable PCB design guidelines are not considered further here.

The presence and details of these factors varies by application and is discussed for the reference application in Chapter 7.

3.5.3 Solder Application Process

Section 3.4.3 discusses the importance of selecting a suitable flux and process parameters for the application of Field's Alloy for achieving good coverage of the low melting point alloy to the carrier PCB. Considering that the implicit goal of designing a module for a self-reconfiguring modular robot system is the manufacturing of a large number of modules, the efficiency of the manufacturing process of soldering connectors also needs to be considered.

An initial consideration was to deposit small quantities individually to each connector pad on the PCB. This creates the problem of finding a suitable method to break the Field's Alloy ingots down into small quantities. Mechanical separation into small spheres was performed in a hot water bath, albeit with little control over the volume of the resulting spherical pieces. Attempts at efficiently implementing the same process with 3D-printed molds failed due to the high surface tension of Field's Alloy. An attempt to use Field's Alloy chips created by coarsely filing the ingot returned sufficiently small pieces to have good control over the amount deposited on each PCB pad, but developing a process for efficiently achieving the same for 650 PCBs with 16 pads each



(a) (b) (c)

(d) (e) (f)

Figure 3.8: Photographs of process steps for applying solder to the soldering connector PCB. (a) Dipping tinned circuit board connector pads in flux. (b) Applying flux manually using a cotton swab. (c) Dipping the tinned pads in hot Field's Alloy. (d) PCB pads after Field's Alloy application. (e) Washing off of flux in warm water. (f) Air drying of PCB.

is a significant engineering problem in itself. Another attractive method considered was to use a heated syringe deposition head on a gantry platform, such as a fused deposition modeling 3D printer, to deposit molten Field's Alloy. This is likely to be the most efficient method for depositing Field's Alloy individually onto pads of very large numbers of soldering connector PCBs, especially when the deposition of flux is implemented on the same platform, for example with a second extrusion head.

(a) (b) (c)

Figure 3.9: Corrosion on soldering connectors as a result of insufficient removal of flux residue. (a) and (b) Two examples of corrosion seen on soldering connector boards where the final flux cleaning step was omitted. (c) Connector PCB with spacer. During maintenance flux was re-applied in an attempt to clean the connector pad and subsequently not thoroughly cleaned, resulting in ignition of the flux-wetted spacer (top left).

The alternative to applying the Field's Alloy individually to each PCB pad is to apply it in bulk to all pads on one PCB in one step. This is the approach ultimately chosen for the 650 soldering connections of the reference application and used for the experiment described above in Section 3.4.3. Figure 3.8 illustrates the steps required to manually apply Field's Alloy to one PCB. First, flux of type Superflux 75 is applied by dipping (Figure 3.8(a)) or using a cotton swab (Figure 3.8(b)). After a 30 s wait the PCB is briefly dipped into Field's Alloy heated to approximately 150 °C to 200 °C (Figure 3.8(c)) resulting in Field's Alloy to be applied to the PCB pads (Figure 3.8(d)). Mild sputtering of flux can occur during this step as the flux is activated only once it is already in contact with the Field's Alloy. Finally, the flux residue is cleaned with warm water (Figure 3.8(e)) and the PCB may be dried with air (Figure 3.8(f)).

The final cleaning step is essential to remove excess flux. If not carried out carefully, corrosion will develop on the soldering connectors after several weeks to several months. Figure 3.9 shows three examples of soldering connector PCBs for which the

flux cleaning step was omitted. The thin layer of white corroded material does prevent two adjacent Field's Alloy pads from fusing effectively making it impossible to form a connection between two thoroughly corroded PCBs. Furthermore, in two cases where flux was applied to re-apply Field's Alloy to connector pads that had been damaged, soaking of flux into the already applied led to the fiberglass spacer's ignition (Figure 3.9(c), the spacer is discussed in Section 3.5.5).

3.5.4 Connector Pad Shape and Error Tolerance

The tolerance to rotational and translational error in the alignment between neighboring modules is directly affected by the smallest distance between Field's Alloy covered connector pads on the soldering connector PCB. Simple trigonometry yields that the minimal distance d_s between two connector pads to tolerate a combined angular error of magnitude ϵ and translational error e, as a function of radial distance r from the center of the PCB, is

$$d_s(r) = rcos(\epsilon) + e \tag{3.4}$$

The amount of translational error tolerated by the connector equals the minimal space between connector pads. A circular pad shape reduces the tolerance to combined rotational and translational error, effectively minimizing the required minimal spacing between pads. Section 7.2.3 discusses methods through which misalignment between modules can be prevented through module design features external to the connector. Another shape for the Field's Alloy covered connector pads that shares the benefit of circular pads is that of concentric annular rings, which has not been explored in depth as part of the work described here but was partially implemented in the prototype shown

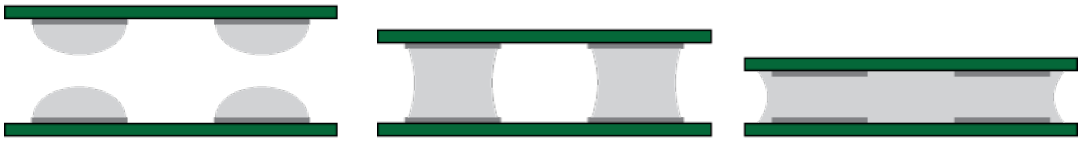


Figure 3.10: Effect of spacing on the connection formed with the soldering connector. (a) If spaced to far apart, Field's Alloy from two adjacent connectors will not touch and no connection is formed. (b) Under optimal spacing, the Field's Alloy on all adjacent connectors' pads touches during the connection process and forms soldered connections. (c) Too small spacing between adjacent connector results in Field's Alloy spilling over the connector pad and potentially forming short circuits across multiple pads.

in Figure 3.1.

Since the symmetry requirement for rotation-invariant connectors implies that there are twice as many pads of those types not placed on symmetry axes as compared to those types on the axes of symmetry, one might consider having pads of different sizes. However, because the contact angle of the Field's Alloy on each pad is approximately the same for all pads, smaller pads have a lower height buildup of Field's Alloy which can prevent them from forming stable and reliable bonds when soldered connections are formed.

The reference PCB design shown in Figure 3.7 includes the design error of placing vias not covered by soldermask close to Field's Alloy pads at different voltage levels. During testing this occasionally lead to short circuits where forming connections led to Field's Alloy spilling over the edge of the connector pad onto the exposed via.

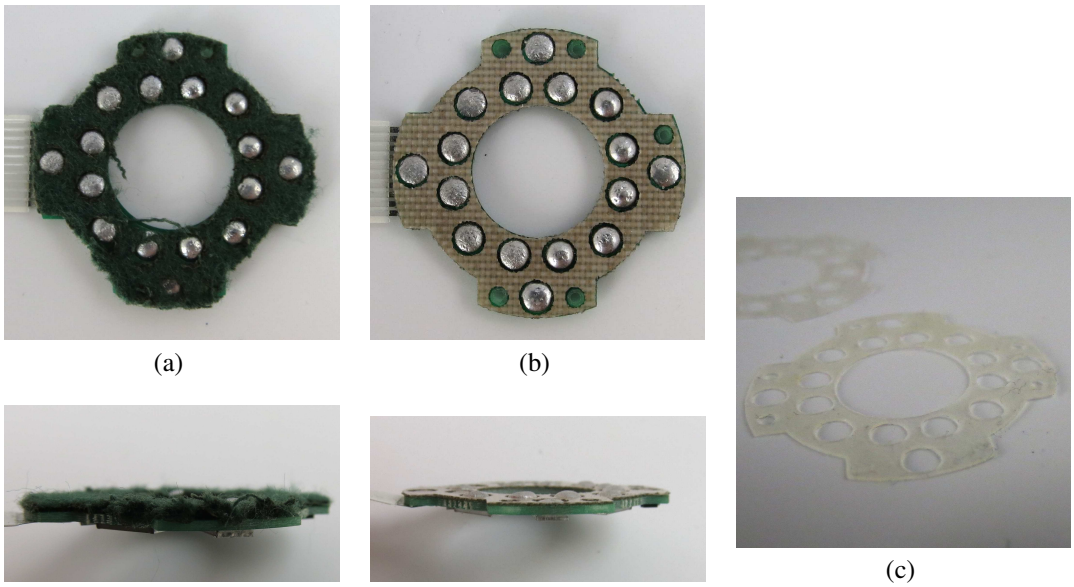
3.5.5 Connector Spacing

The spacing between two adjacent connectors is critical to forming a functioning electrical connection between the two connectors. If the spacing is too large no connection is formed at all (Figure 3.10(a)), but if the spacing is too low fields metal might be squished between the two connectors and form unintentional connections or short circuits with neighboring pads.

In order to ensure optimal spacing between adjacent connectors, a spacer is applied to the exterior facing side of the connector PCB, which holds the Field's Alloy. Use of an Object 3D-printed spacer as shown in Figure 3.11(c) has the benefit of giving free choice over the thickness of the connector in 16 µm increments - the printer's z-axis resolution. However, the Objet compatible materials investigated deformed under heating and application of an adhesive to the spacer but not the Field's Alloy proved difficult. Alternatives available commercially off-the-shelve are various adhesive backed materials including adhesive backed felt (Figure 3.11(a)) and PTFE-coated fiberglass film (Figure 3.11(b)). Felt proved difficult to laser cut and thus the fiberglass film was chosen as spacer material for the reference application. Additionally, the PTFE coating of the fiberglass film provides for a low friction non-stick outward facing surface. To expose the Field's Alloy, the stencil layer of the PCB as shown in Figure 3.7(e) is laser cut into the adhesive backed films that have a thickness of 0.010 in (0.254 mm).

3.6 Experimental Validation

After Sections 3.4 and 3.5 discussed the design of the self-soldering connector for modular self-reconfiguring robots, the following describes several experiments that aim to



(a) (b) (c) (d) (e)

Figure 3.11: Soldering connector spacer implementations. (a) and (c): Adhesive laser cut felt, (b) and (d): Silicon-adhesive laser cut fiberglass film, (c) 3D-printed.

validate the suitability of the connector. The previous sections made a distinction between discussing the general design process for creating a self-soldering connector for any application, and the reference design developed in the context of an application that will be discussed later in this thesis in Chapter 7. To experimentally validate the design concept we must now rely on specific implementations of the soldering connector, because it is of course not possible to perform experiments on a generalized concept. This is why Section 3.6.1 first introduces three implementations of the self-soldering connector with references to other parts of this thesis for in-depth discussions of their application, before Sections 3.6.2 to 3.6.4 discuss tensile test experiments and a repeatability study respectively.

3.6.1 Soldering Connector Implementations

10 Millimeter Sized Connector

The 10 mm square connector design has been designed as a prototype for the demonstration of stochastic fluidic self-assembly described in Chapter 6. The PCB of the connector is sized 10 mm by 10 mm and six of these PCBs cover almost the entire surface of a cube shaped module, shown in Figure 6.2. The hole at the center of each PCB is the port of a channel structure in the interior of the module through which fluid flow can be directed when the module is part of an assembly. The Field's Alloy covered connector pads are arranged around the circumference of the PCB with a spacing of 1 mm to the edge. Each pad measures 1 mm^2.

15 Millimeter Sized Connector

In a later version of the stochastic fluidic self-assembly system the connector design is updated to approximately double the surface area to 15 mm^2 and reduces the number of distinct electrical lines passed through the connector to two.

1 inch Sized Connector

The reference design of Section 3.5 was implemented in the context of a 55 mm cube shaped module of a self-reconfiguring modular robot system. The system and demonstrations performed with this modular robot system are described in detail in Chapter 7. The surface area available for the soldering connector in this system is approximately 1 in^2 and the final connector PCB design is shown above in Figure 3.7 as drawings and in Figures 3.11(b) and 3.11(e) in a photograph with the attached spacer. The connector

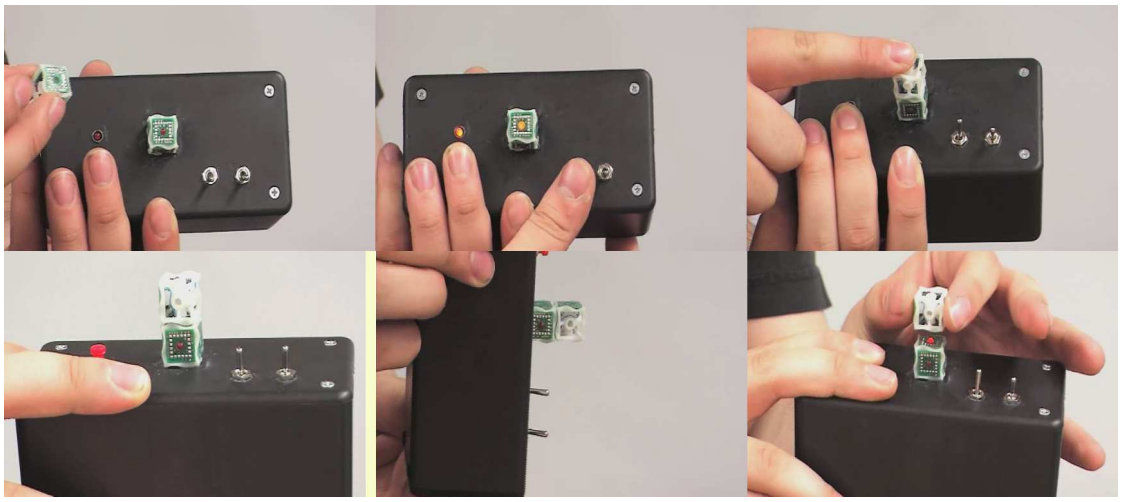


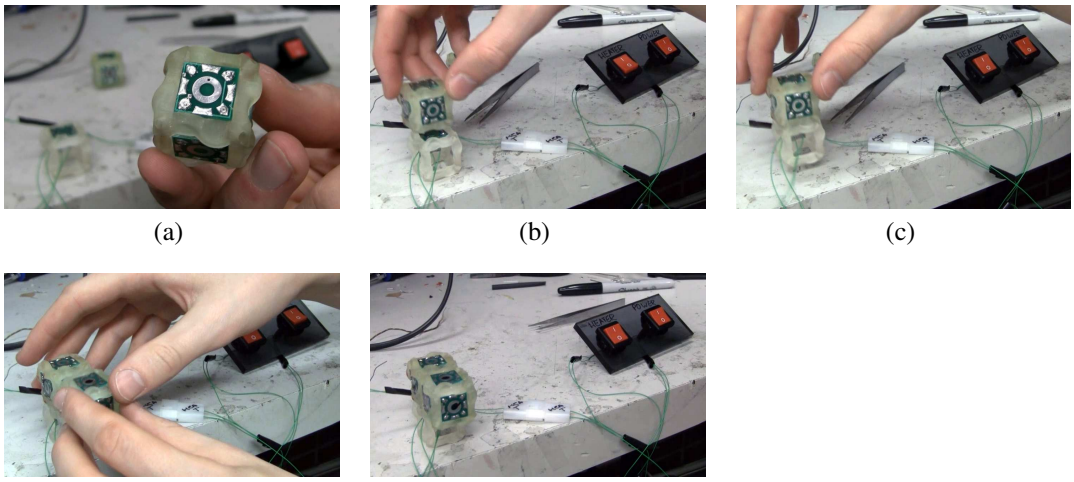
Figure 3.12: Proof of concept of soldering connection method as mechanical connection. Top left to bottom right: A cube is connected to a substrate, and it's upper connector PCB is heated as indicated by the embedded LED. After 20 s of heating, a second cube is attached and held in place manually for 60 s with heating stopped. The cube does not fall with gravity (and supports the weight of the substrate, not shown) suggesting that a mechanical connection has been formed. After another round of heating, the second module can be easily disconnected.

PCB has 16 3.0 mm diameter pads covered in Field's Alloy.

3.6.2 Proof of Concept Demonstrations

Proof of Mechanical Connection with 10 mm Sized Connector

To demonstrate the feasibility of modules forming a self-soldered connection to form a mechanical connection, a simple proof-of-concept demonstration with two modules was developed, shown in Figure 3.12. In this demonstration one cube is permanently connected to a substrate that houses a battery, and to two switches to control heating. Another passive cube is connected to the first one. The fixed module's upper connector PCB is heated, as indicated by an LED embedded for visualization purposes. After



(a) (b) (c)

(d) (e)

Figure 3.13: Proof of concept of soldering connection method as electrical connection. (a) A module with six soldering connectors and all power supply pins and ground pins interally connected. (b)-(c) The module is attached to a substrate with heated soldering connector. (d)-(e) After supplying current through the newly formed connection, the recently attached cube can now heat its soldering connectors and form a connection with a third cube.

20 s of heating, the second cube is attached and held in place manually for 60 s while heating is stopped. To test whether a mechanical connection exists, the assembly is first held upside down, and secondly the attached individual module is used to lift the entire assembly. In both cases the connection does not break, suggesting that a mechanical connection has been formed. After heating the fixed module's top connector PCB again for 20 s the second module can be easily disconnected, suggesting that the connection is indeed reversible.

Proof of Electrical Connection with 15 mm **Sized Connector**

Using the larger version of the self-soldering connector, a further experiment involving three modules was devised to demonstrate that in addition to the mechanical connection an electrical connection is formed. Figure 3.13 shows a sequence of steps involved

in this demonstration. As before, one module acts as substrate and is connected to a power supply and to switches controlling whether the substrate module's top connector is heated and whether voltage is applied to its connector pads. After a few seconds of heating, a first module is attached to the substrate module, and after approximately one minute of cooling it is confirmed that a mechanical connection has formed by lightly attempting to pull the two modules apart.

To confirm a correct electrical connection, voltage is now applied to the substrate module. No short circuit is detected, confirming that no unintended electrical connections between connector pads have been formed. Inside the newly connected module, the connector pads of all of its six connectors are electrically connected to pass voltage applied from any neighboring module to all others, as would be the case in the completely functional module where the communication and power supply lines are connected in the same fashion. Further, this module is modified from the final design such that whenever voltage is applied, all six connector PCBs' heaters are enabled. This allows a third module to be attached. After a minute of cooling, it is again confirmed that a mechanical connection exists by lightly pulling on the assembled structure, and the absence of short circuits suggests that no incorrect electrical connection was formed.

The absence of a spacer as discussed in Section 3.5.5 is justified in this prototype design by the wide edge of the module shell into which the connector PCBs are embedded. The design of the shell of this module is such that the PCB is recessed into the shell ensuring sufficient spacing between the two connectors of neighboring modules.

With the feasibility of forming mechanical and electrical connections using the soldering connector established, the usefulness of those connections must now be established in the context of the requirements for self-reconfiguring modular robot systems outlined in Section 3.1 above. Several requirements are addressed by the basic con-

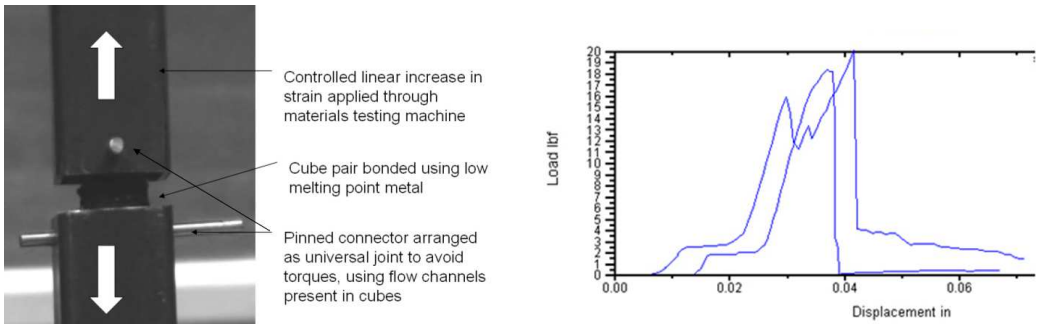
cept of the soldering connector: Information and power transmission capability as well as non-continuous power consumption are inherent to the design of the soldering connector, as is reversibility, which has also been demonstrated by the proof-of-concept demonstrations in this section. Small size, tolerance to error, the magnitude of power consumption, and rotation invariance are achievable by appropriate design of the PCB layout for the soldering connector, and are addressed in the System Design section above (Section 3.5). The remaining requirements, mechanical strength and repeatability, are addressed by experiments in the following two sections.

3.6.3 Tensile Tests

The tensile strength of the connection between two modules in a modular robot system directly affects the strength of the entire modular structure. For basic operations in a modular assembly the forces encountered from supporting other modules, which might be hanging or cantilevered, must be supported. In any real system forces will also be caused by small misalignment and friction between modules when moving. Even though it is difficult to predict the exact forces and torques that might be acting on a connector during normal operation, tensile tests will give an indication of the magnitude of loads the connector can support, and are useful for comparing different connector types.

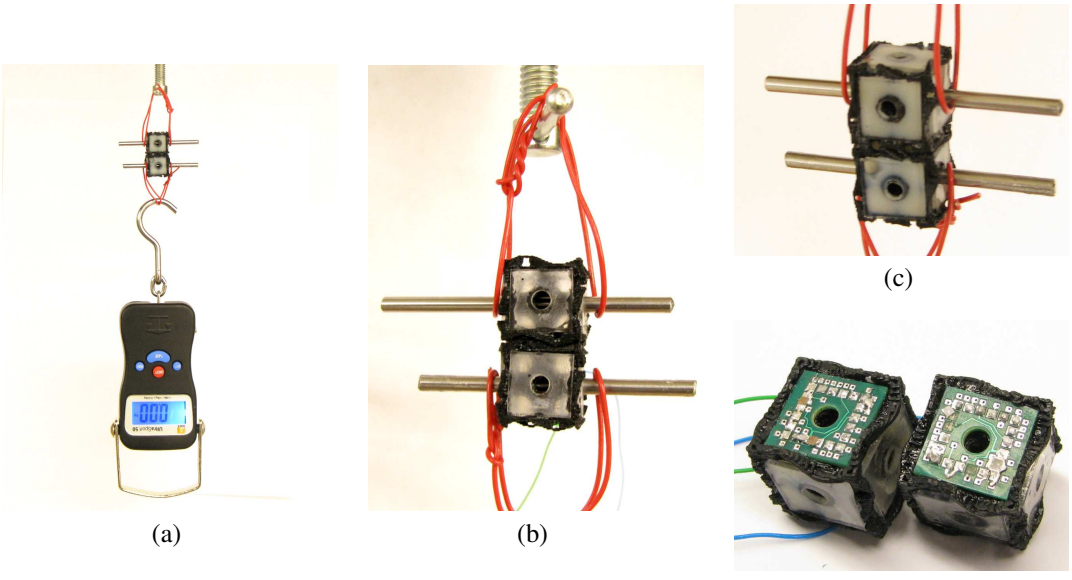
Experiment with 10 mm Size Connector

A first tensile test was carried out using a prototype module with the $10\,\mathrm{mm}^2$ size connector. Field's Alloy was applied to the connector pads of two connector PCBs, which where then brought into contact and externally heated with hot air. Once fused the modules where embedded in module shells using adhesives. The module shell was 3D-



(a) (b)

Figure 3.14: Tensile test of two prototype modules with the 10 mm size connector in an Instron Material Testing machine. (a) The mechanical setup of the test. (b) Results for two tensile tests, recorded from the tensile test machine as screenshot.



(a) (b) (c) (d)

Figure 3.15: Simple setup for validating tensile testing results. (a) Two prototype modules with the 10 mm size connector in tension and a connected scale. (b) Closeup of (a) showing the cube with simple adhesive mounted PCBs. (c) Same as (b) but with epoxy-filled module interior to prevent module breakage. (d) Connector surface after failure in tension with epoxy-filled module interior.

printed in the fused deposition modeling technique on a Stratasys Dimension printer, and the adhesives used were Superglue® and Loctite® Hysol E00-L fast setting epoxy.

To connect the two modules to an Instron material testing machine a custom mounting shown in Figure 3.14(a) was manufactured. The smallest available load load cell with a maximal load of 1000 lbf load cell was used and the machine set to elongate the sample at a speed of $1\,\mathrm{mm\,min^{-1}}$. Figure 3.14(b) shows the test results for two experiments in the format presented by the instrument. The maximal loads of 18 lbf and 20 lbf (80 N and 89 N) suggest that the connection can support several kg. However, it should be noted that in both tests, the point of failure invariably was in the mounting method of the connector PCBs, and not in the solder connection between two connected PCBs: Instead of the two connector PCBs coming apart during the test, the module shells broke.

With limited access to the Instron material testing machine, followup experiments were conducted using a much less precise setup: Load was applied in small increments to the pair of modules through a fishing scale. The readout of the scale was monitored using a video camera and after failure the last readout before failure was read from the recorded video. Figure 3.15(a) shows the overall test setup, Figure 3.15(b) shows a closeup image of the pair of modules where Superglue® was used as adhesive to hold the circuit boards in the module shell, and Figure 3.15(c) shows the same setup but with the entire module filled with Loctite® Hysol E00-L epoxy to prevent the module itself from breaking. One test was observed where the two connector PCBs separated and the resulting PCB surfaces are shown in Figure 3.15(d). Notable features are:

- Several connector pads are electrically connected with Field's Alloy bridging them. In addition, Field's Alloy has spilled from the pads onto adjacent vias. This suggests that spacing between pads, or spacing between the two PCBs, or both need to be adjusted for this connector design.

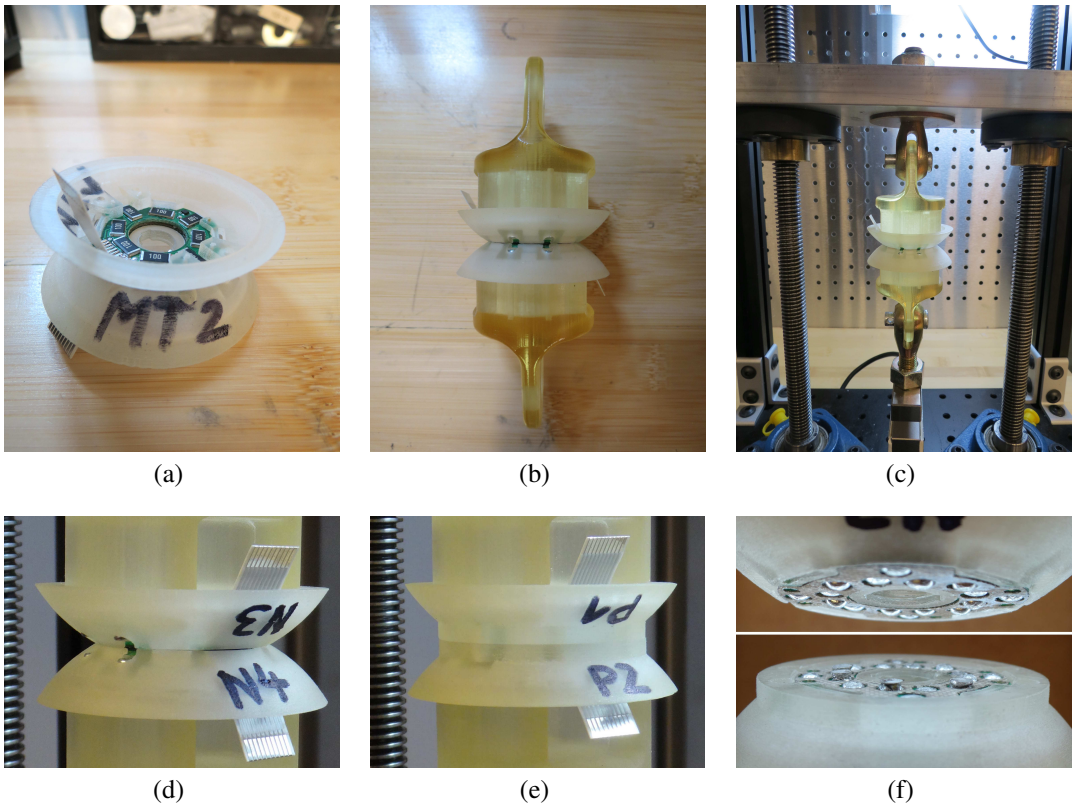
- Some connector pads' Field's Alloy cover exhibit a fractured surface, suggesting that the solder connection broke in half.
- On other connector pads no fracture surface is visible, suggesting they might not have connected to their respective neighbor on the adjacent PCB at all.
- On some connector pads the tinned copper pad has disconnected from one connector's PCB and is still soldered to the pad of the other connector. For these pads, the failure has not been breakage of the soldered connection, but the connection between the PCB's copper layer and the PCB substrate.

Experiment with 1 in **Sized Connector**

Given the observations of the tensile tests with the 10 mm^2 size connector, several design iterations of the soldering connector were investigated for several application scenarios, ultimately resulting in a PCB design for the 1 in^2 size connector. A series of tensile tests was performed on connectors of this design using the *freeLoader* test apparatus described in [152].

During the time between the tests on the 10 mm^2 size connector and the new tests described here, most aspects of the connector design including the size, shape, and arrangement of the connection pads as well as the Field's Alloy application method were updated, resulting in a completely overhauled design. Given the changes in the connector design and the use of a different test apparatus, a direct comparison between the two series of tensile tests is not meaningful.

Figure 3.16 shows photographs of the test setup: Connector PCBs are fastened into partial shells of the modular robot modules described in Chapter 7 using miniature screws and adhesive. The use of a partial shell allows for access to the heater side



(a) (b) (c)

(d) (e) (f)

Figure 3.16: Tensile test of of pairs of 1 in sized connector in freeLoader tensile testing apparatus. (a) Connector PCBs are inserted into partial shells of robot modules leaving the heater side of the PCB accessible. (b) The partial module shells with attached connector PCBs are attached to 3D-printed adapter jigs. (c) Using the jigs the pairs of connected connector PCBs are mounted in the freeLoader material testing apparatus. (d)-(f) Difference between the regular module shell with 1.2 mm spacing between connected PCBs in (d) and top in (f), and the PCB with added spacing resulting in a total gap of 1.8 mm between connected PCBs in (e) and bottom in (f).

of the connector PCB during test preparation when the two connectors are joined. The joining process is identical to how two complete modular robot modules would connect. The power supply and control signal that would normally be sent by a module controller simulated by external power supplies. The partial shells with embedded connector PCBs are then attached to 3D-printed adapters, shown in Figure 3.16(b) in oder to be mounted into the freeLoader material testing apparatus, shown in Figure 3.16(c).

Table 3.6: Tensile Test Batches and Test Parameters

Batch	Elongation Rate	PCB Spacing
Batch 1	$3\,\mathrm{mm\,s^{-1}}$	1.2 mm
Batch 2	$3\,\mathrm{mm\,s^{-1}}$	1.8 mm
Batch 3	$2\,\mathrm{mm\,s^{-1}}$	1.2 mm
Batch 4	$2\,\mathrm{mm\,s^{-1}}$	1.8 mm

A total of twenty tensile tests were carried out, grouped into four batches. For each test, the pair of connectors were joined as described above with current applied to the heaters until a temperature of 90 °C was measured on the heater surface using a thermal imaging camera. The 90 °C was chosen arbitrarily to ensure repeatability in the tensile test setup. After heating to form a connection the connected pair of connector PCBs were let stand to cool for five minutes and until the heater surface temperature was confirmed to be below 35 °C. Subsequently, the pair was mounted into the tensile testing machine and the tensile test started.

Ten tensile tests were carried out at a rate of elongation of $3\,\mathrm{mm\,min^{-1}}$, and ten others at $2\,\mathrm{mm\,min^{-1}}$. This is to investigate if there exist time dependencies in the material behavior. Table 3.6 shows the association between batches and test parameters.

The second test parameter investigated is spacing of connector PCBs in the test: Ten tests were performed with the regular module shell which has a spacing of 1.2 mm between the copper layers of the connected PCBs, and ten tests performed with shells that have a 0.3 mm standoff resulting in a total gap of 1.8 mm between two connected PCBs. A pair of connectors embedded in a cube shell that results in 1.2 mm is shown in Figure 3.16(d), and a pair resulting in 1.8 mm spacing is shown in Figure 3.16(e). Figure 3.16(f) shows individual connector PCBs embedded in the two respective shells, with regular spacing (top) and 0.3 mm standoff (bottom).

Four test batches with five tests each were performed covering each permutation

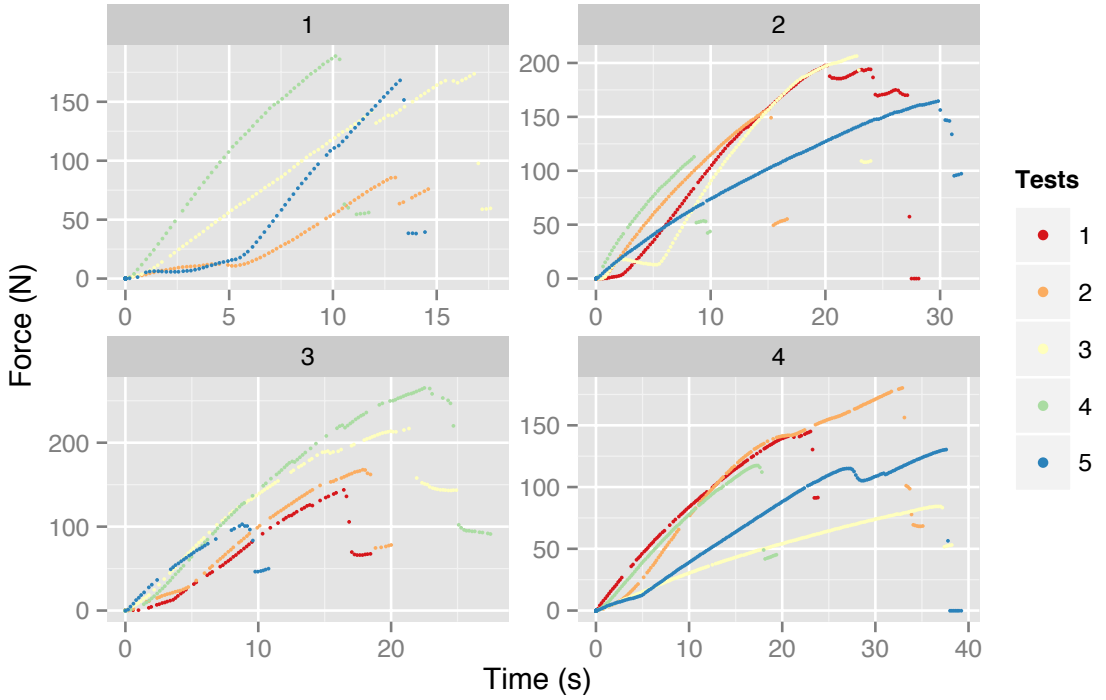


Figure 3.17: Time history of force measured from tensile tests on pairs of 1 in^2 size connectors on a freeLoader tensile test apparatus.

of the two test parameters, rate of elongation and spacing between connectors. Out of those, one test result was excluded from the result analysis because the adapter jig failed. Figure 3.17 shows the force-strain curves obtained for 19 tensile tests, split by their batch and color coded by tests 1 through 5 for each group. The results are presented in terms of force, as opposed to stress, because not all of the redundant connector pads always form a connection, as we will see below. Thus we do not know the surface area of the connection and stress cannot be calculated.

The average maximal tensile force at failure is found to be 173 N (39 lbf) with a standard deviation of 46.4 N and the average elongation at failure is 0.89 mm with a standard deviation of 0.55 mm. The large standard deviation of the elongation at failure is explained by the two error modes captured in the experiments: While in most experiments the soldered connections failed, two tests led to the connector being pulled out of

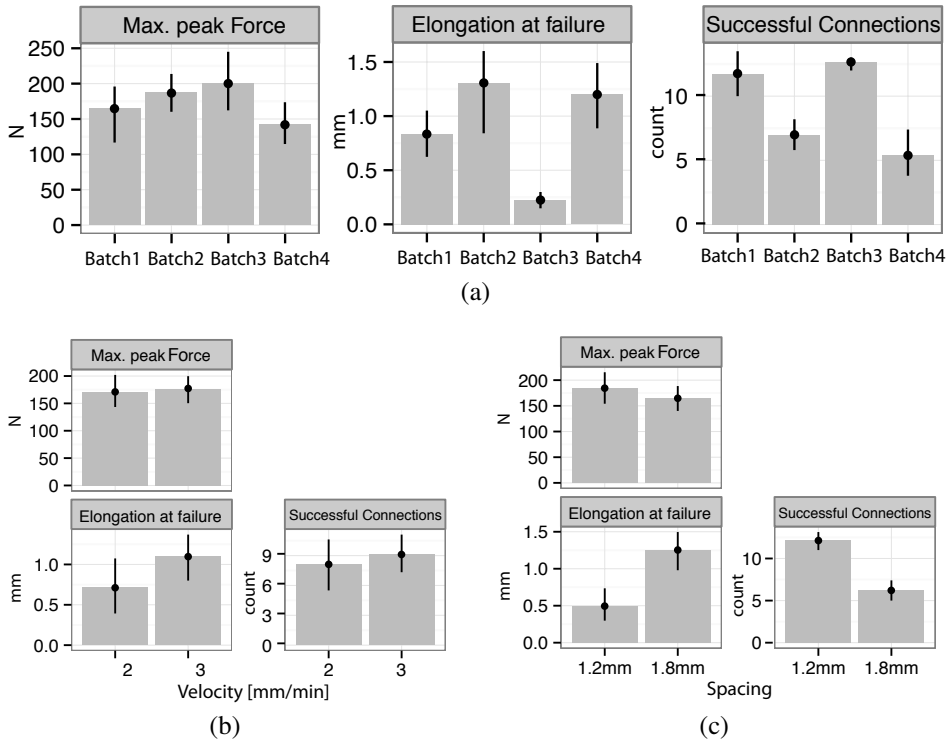


Figure 3.18: Results for tensile tests on pairs of $1\,\text{in}^2$ size connectors on a freeLoader tensile test apparatus. (a) Per batch with parameters described in Table 3.6. (b) Aggregated by elongation rate. (c) Aggregated by initial connector PCB spacing.

the partial module shell.

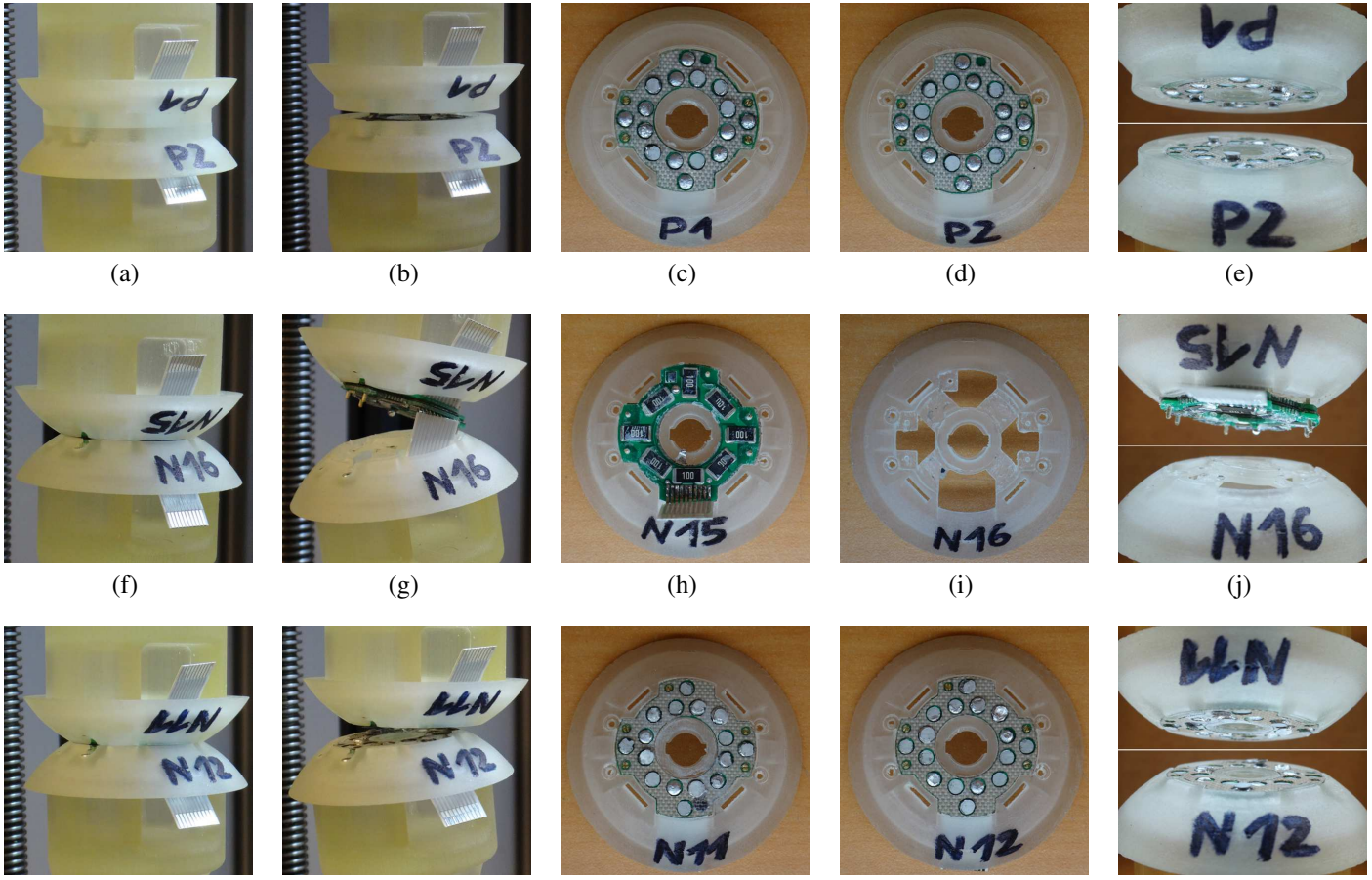
Figure 3.18 helps understand the impact of elongation rate and spacing on the test outcome. In addition to the force and elongation at failure, the number of broken connector pads is recorded. Broken connector pads are those where a fracture surface is clearly visible or where the copper pad was disconnected from one of the PCBs during the failure of the connection. Connector pads that show neither of these features did not form a soldered connection during the connection process, and did therefore not carry any load or break during failure. Figure 3.18(a) displays average and standard deviation of the metrics described above per batch, while Figures 3.18(b) and 3.18(c) aggregate the same results by elongation rate and connector spacing respectively.

Grouping the results by elongation rate does not show any statistically significant difference suggesting that there is no time dependence in the failure modes observed. Grouping by initial spacing between the two connector PCBs does, however show a significant difference in the observed metrics. Most notably, the number of pads that form a soldered connection with their respective counterparts on the other connector is only 50 % of the value found for the smaller spacing. Surprisingly, this does not cause the force at failure to be reduced by the same factor, suggesting that in all cases only a small number of individual solder joints contributes to the ability of the soldered connection to support loads.

Figure 3.19 shows time series photos of three tensile tests with the connectors before test start, shortly before or just after failure, as well as top and side views of the connector after failure. Figures 3.19(a) to 3.19(e) show a test where only five out of the 16 connector pads had formed a solder joint, Figures 3.19(f) to 3.19(j) show a test where the connection between PCB and module shell broke before the solder joints, and Figures 3.19(k) to 3.19(o) show a test where all 16 solder pads had formed a soldered connection.

3.6.4 Repeatability Test

To evaluate the durability of the connection method of the self-soldering connector, repeated connection-disconnection cycles were performed in an automated experiment using a CRS A465 robot arm. The test setup, shown in 3.20, consists of a 3D-printed partial module shell mounted to the robot arm in place of an end effecter, and a second similar shell mounted to the work surface. Connector PCBs are inserted into both partial shells using adhesive and miniature screws to result in a connector that is identical to



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

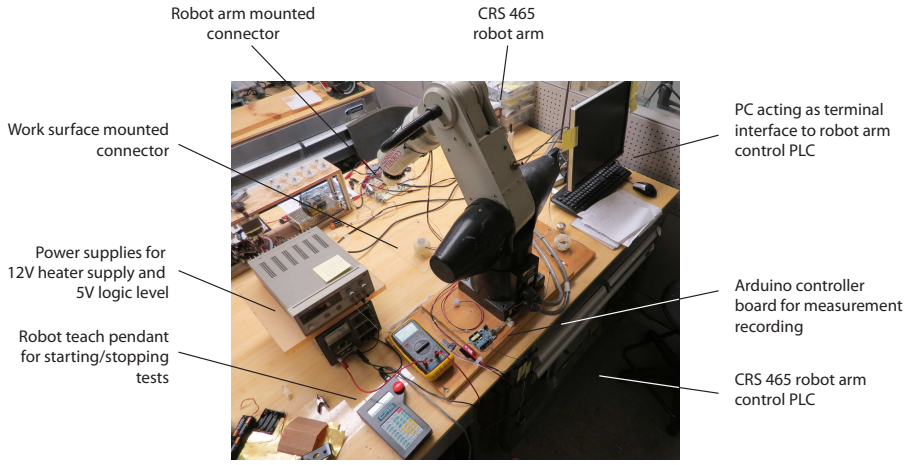
Figure 3.19: Photographs of failure modes observed during tensile tests of pairs of soldered connectors. (a)-(e) Only 5 out of 16 connector pads show a fracture surface suggesting an incomplete connection was formed. (f)-(j) One connector was pulled out of the cube shell. (k)-(o) All 16 connector pads show fracture surface suggesting a complete connection was formed.

those found on the robot modules described in Chapter 7 of this thesis. While the lower connector is fixed to the work surface, the connector attached to the robot arm is spring mounted and can travel up and down in the z-direction on a low friction lubricated 3D-printed slider. This is to simulate the flexibility inherent to a system of 3D-printed modules when compared to the rigidity of a robot arm.

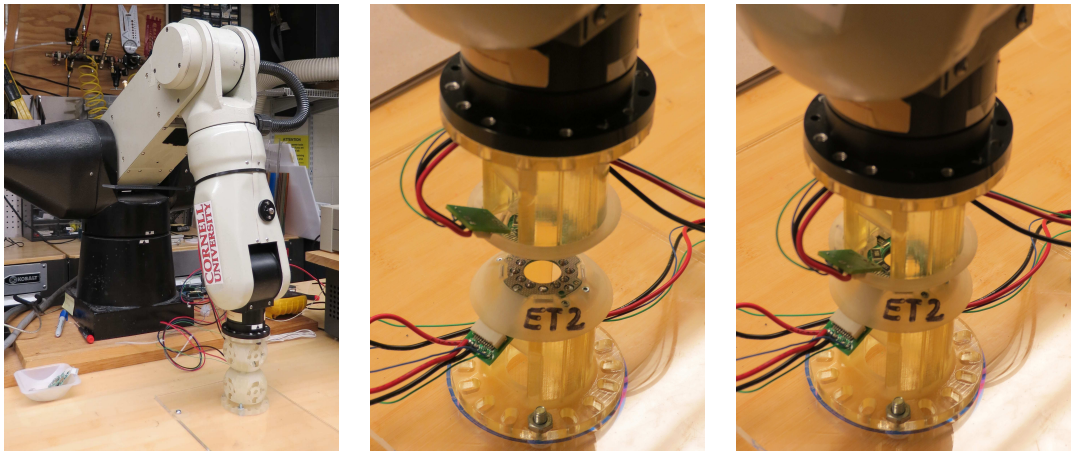
The connectors in the test are electrically connected in a fashion identical to connectors in a complete robot system. Both connectors are connected to a 12 V power supply that both is used for the soldering connector's heaters and applied to the connector pads for power distribution through the connector. The logical level control line of the switch, which controls the soldering connector's heaters, is wired to a digital output of the CRS A465 robot arm's PLC controller and is controlled from the same program that controls the movement of the robot arm. The digital output rail of the CRS A465 controller is externally supplied with 5 V to make it compatible with the soldering connector's switch.

The communication signal lines exposed on four pins of each soldering connector, are connected to digital input/output pins of an Arduino™ Duemilanove microcontroller board. The Arduino control program switches either the robot arm mounted connector to be connected to an Arduino output pin, and the table mounted connector to be connected to an input pin, or vice versa. This setup is used to determine whether the communication pads of the two connectors are electrically connected: The Arduino sends a signal to one connector and checks if the same signal is received from the other connector.

One failure mode of the connector are unintended connections between the different voltage levels exposed on the Field's Alloy covered pads of the soldering connector. Several methods are in place to detect these: The 12 V power supply line is connected through a potential divider to a 5 V digital input pin of the Arduino board. When a



(a)



(b) (c) (d)

Figure 3.20: Automated connection repeatability test: setup. (a) System overview of test setup. (b) Front view of CRS 456 robot arm while connecting two connectors. (c) Connectors before connecting. (d) Connectors when brought in contact by robot arm.

short circuit between power supply and ground occurs, the input of the Arduino board will turn from logical true to false. Other unintended connection could occur between the communication line and either 12 V or ground. To detect these, it is important to check for both the correct transmission of both high and low signal voltages: If one were to only check if the signal line is high when set to be high, a short circuit between the power supply line and the signal line would result in a positive reading. Each test

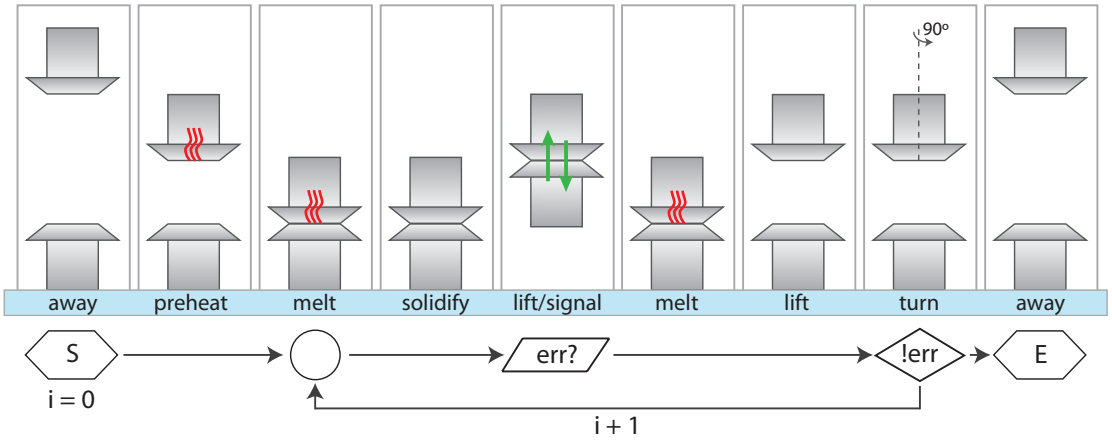
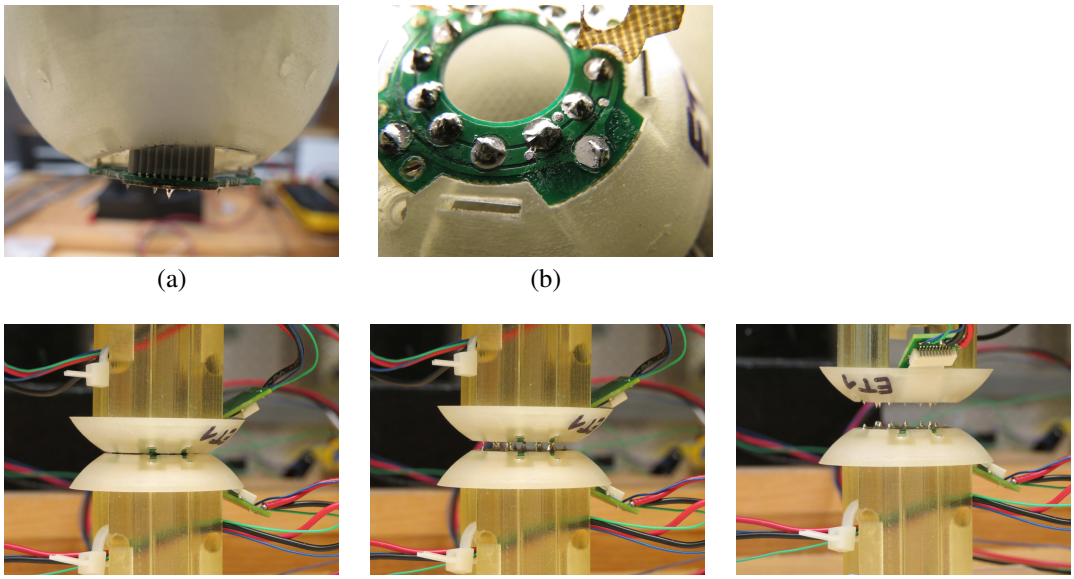


Figure 3.21: Sequence of actions during one test cycle in the repeatability test.

of the transmission line involves five 200 ms high pulses alternating with 200 ms low periods. All five high and low signals must be detected at the receiving connector for the connection to be valid.

The Arduino board is connected to a PC through its USB connection over which logs including the results of tests on the communication line and the state of the power supply line are sent in serial format. The Arduino is programmed to send a test signal through the connector's communication line, as described above, and read the status of the power supply line upon receipt of a trigger pulse on one of its digital input pins. This trigger pin is wired to a digital output of the CRS A465 robot arm's PLC controller in the same way as the heater switch. This makes the robot arm's control program, which is written in a proprietary robot control software and executed in the robot arm's PLC controller, the central point of control over the test workflow.

The workflow for each experiment is visualized in Figure 3.21: Starting from an "away" position the experiment starts with a short 5 s preheating phase before it enters a cycle of a user defined number n connect-disconnect disconnect cycles. After the connection is formed, the robot arm raises to it's "disconnect" position and if a soldered connection exists, the lower connector is lifted and the spring holding it in place is



(a) (b)

(c) (d) (e)

Figure 3.22: Automated repeatability test results. (a) Failure mode of PCB connector being pulled out of the module shell. (b) Short circuit, the Field's Alloy on the connector pad in the front of the picture has spilled off the connector onto a nearby exposed via, exposing the presence of exposed vias as design flaw of the connector implementation. (c)-(e) Side view of two connectors disconnecting. The remaining "spikes" settle back into dome shape if the connector is kept heated for a few seconds after disconnection.

stretched applying a force to the connection. During this state, the trigger pulse for testing the electrical soundness of the connection is sent to the Arduino board.

During preliminary tests several runs with hundreds of successful cycles were achieved, including one test that was terminated after 1075 cycles without failure. To make the test more realistic, a 90 degree rotation of the robot arm mounted connector after each cycle was added, taking into account the assumption that during normal operation in a modular robot system repeated connections between the same two connectors in the same relative orientation are unlikely.

A difficulty encountered during preliminary testing was to keep the low tempera-

Table 3.7: Repeatability Test Results

Batch	Cycles to Failure	Failure Mode
# 1	70	short circuit
# 2	46	short circuit
# 3	64	short circuit
# 4	502	short circuit
# 5	422	short circuit

ture of the connectors in all cycles constant throughout the test cycles. If not accounted for, insufficient cooling might result in Field's Alloy not solidifying between cycles, or random effects might cause Field's Alloy not to melt resulting in a test failure. This problem was addressed by installing a PC cooling fan next to the two connectors and adjusting the wait times between and during cycles such that the heater surface temperature as measured using a thermal imaging camera always drops below 40 °C during cooling periods and raises above 65 °C during heating periods.

After using tens of preliminary tests to tune the test parameters to support large numbers of test cycles, five repeatability tests were performed under the controlled temperature conditions described above. Their results are listed in Table 3.7, with the mean number of cycles to failure being 221 at a standard deviation of 222 cycles. The only failure mode encountered was short circuits between the power supply line and ground with no failure of signal transmission on the communication line. This is likely explained by the arrangement of connector pads which is such that signal transmission connector pads are surrounded by larger margins than power and ground pads. One cause of short circuits are cases where Field's Alloy spills on exposed vias in the PCB design as illustrated by one sample in Figure 3.22(b), exposing the presence of exposed vias, particularly those placed close to connector pads, as a flaw in this implementation of the soldering connector. The other cause of short circuits is Field's Alloy forming connections between connector pads of different voltage level, as is visible in several of

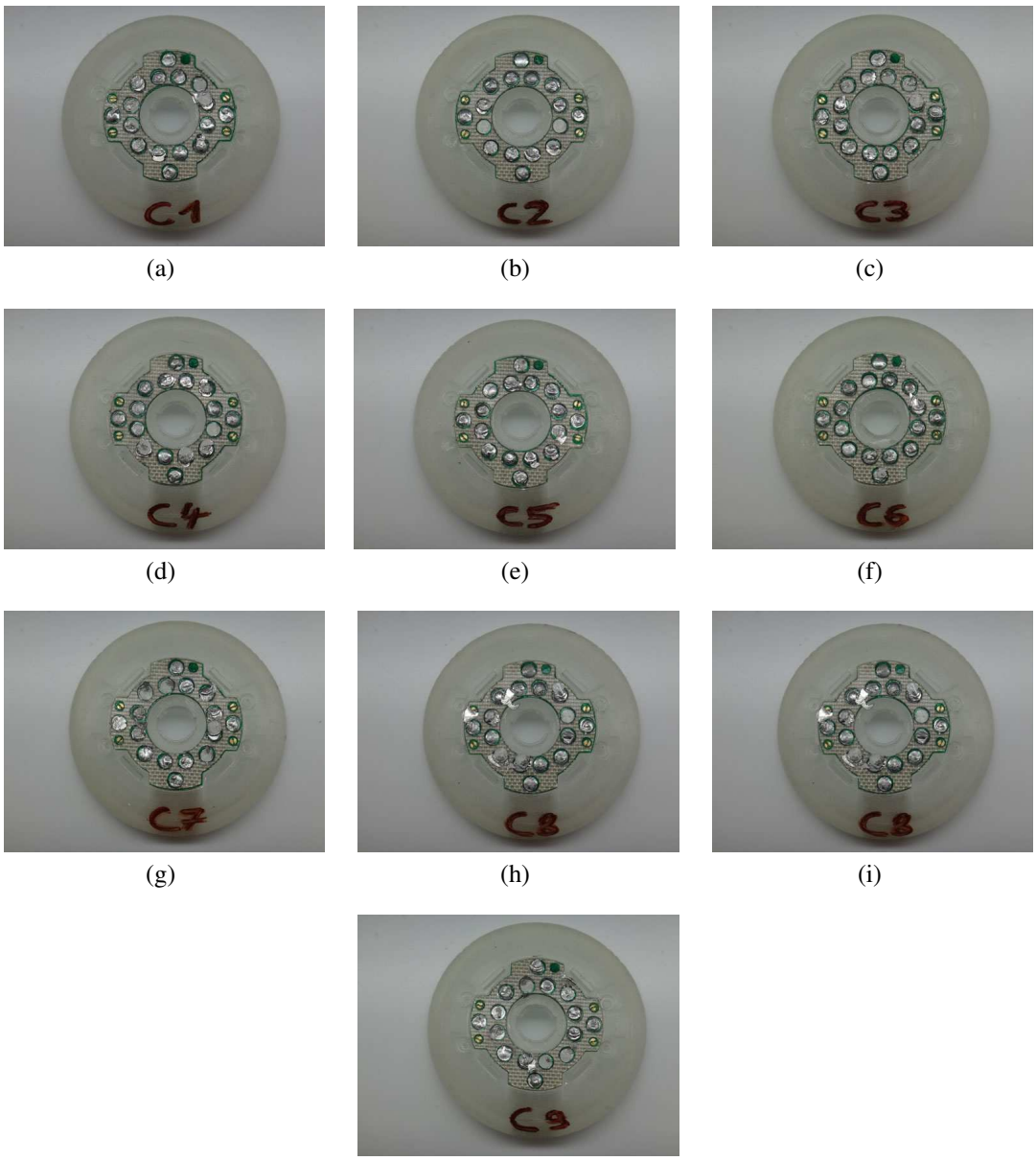
the top views of connectors after failure in Figure 3.23. The majority of this second type of short circuits is for cases where all Field's Alloy that was applied to one connector pad has disconnected from the pad, resulting in an excessive amount of Field's Alloy on the pad it last formed a soldered joint with. This error mode might be preventable if a stronger connection between the PCB pads and the applied Field's Alloy could be created. No mechanical failure of the connection was observed during either controlled nor preliminary tests except when the cooling time during the test cycle was kept too short for the Field's Alloy to solidify.

3.7 Discussion

This chapter presented a connector for modular self-reconfiguring robots based on forming solder joints with a low melting point alloy. The connector is suitable for autonomous operation, with no force to be overcome through external manipulation during either connection or disconnection. By selecting an appropriate solder, flux, and heater, a design sized for typical modular robot module designs was achieved that requires approximately 7 W for 10 s during connection or disconnection in air, and can function when submerged in liquids such as oils and distilled water.

What differentiates the self-soldering connector from other connection mechanisms for modular robots are it's very low complexity, weight, size, and cost. The connector has no moving parts, is fully contained on a printed circuit board, contains only cheap surface mount resistors and one MOSFET device for switching, and can easily be manufactured in bulk using the method developed in this chapter.

The self-soldering connector was developed as a general concept and design guidelines are provided for developing application specific implementations. In addition sev-



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j)

Figure 3.23: Top view of connectors after repeatability tests. Grouped in pairs consecutively in order of test performance as listed in Table 3.7.

eral implementations of the self-soldering connector were presented and used for validating the concept. In particular the 1 in sized version of the connector was validated using rigorous tensile test and repeatability experiments. This implementation has a total weight of 2 g per connector and thickness of less than 3 mm allowing for it to be embedded into the shell of the robot module. The tensile load supported by the connector before failure is 173 N or approximately 8800 connector weights. This is weaker than or roughly equal to the connection methods used in other self-reconfiguring modular robot systems in absolute terms. However, when put in relation to the module weight of a modular robot system, systems that integrate the self-soldering connector are likely to exceed thanks to the weight and volume reductions possible due to using the connector.

One caveat of the connector is the variable repeatability. While the mean number of 220 connection-disconnection cycles before failure is sufficient to support experiments typically carried out with modular robots today, the high variability in this number with three out of five tests failing after less than 100 cycles can be problematic for many applications. A comparison to other systems is not possible, because no repeatability test results have been published for other modular robot connectors. Several strategies could be considered for improving reliability of the connector: The implementation used for the repeatability test has the critical design flaw of exposed vias on the the connector PCB which should be avoided, and reducing the size of the connector pads and increasing their spacing is likely to lead to significant increases in the number of cycles to failure. Further, a reduction of the area of connector pads will likely lead to a smaller probability of short circuits, but might reduce the mechanical strength of the connector.

Future work on the self-soldering connector should be directed at improving the repeatability of the connection method and the development of a fully automated assembly

process including the Field's Alloy application. The selection of available fluxes is vast and only a small subset could be explored in this work: Future research in the materials could explore the space of combinations of circuit board copper coatings and fluxes further. Most importantly, however, the design of application specific implementations of the self-soldering connector is encouraged. The connector is self-contained and small enough to be used as a drop-in replacement for existing modular robot systems and is simple to embed into new designs, requiring only a power supply and on control signal. Beyond modular self-reconfiguring robots the self-soldering connector could be useful wherever a reversible connection is required that is both mechanically strong and electrically conductive.

CHAPTER 4

MESOSCALE THERMORHEOLOGICAL VALVES[†]

4.1 Motivation

In stochastic assembly systems unconnected modules move freely in a mixed assembly medium. The assembly process takes place when modules collide and, if compatible, connect. In two dimensions this can be demonstrated by placing tiles whose surfaces are patterned with magnets on a shaking air table: Upon collision, tiles with compatible magnetic connectors will form connections, while others will not.

A *self*-assembly system exists, when the modules dynamically change their connectors to control the assembly process. The *Programmable Parts* system by Bishop et al. and *Stochastic 2D* system by White et al. demonstrate the two dimensional implementation as tiles which can change the magnet pattern on their surface on the fly, by rotating some magnets into opposite polarity [22, 107].

Building a three dimensional demonstration of stochastic self-assembly is challenging. One approach involves to use a liquid as assembly medium and placing the process inside a stirred tank, as demonstrated by White et al. and Zykov et al. in their respective implementations of the *Stochastic 3D* system [23, 125]. The dynamic control over the assembly process here lies with those modules already assembled around a fixed seed. To redirect flow within the assembly chamber, the already assembled structure requires the capability to selectively create flow sinks and sources on its surface. This is achieved by dynamically routing fluid flow through the interior of the already assembled struc-

[†] This chapter includes content previously published in Neubert, J., Cantwell, A., Constantin, S., Kalontarov, M., Erickson, D., Lipson, H. (2010) "A Robotic Module for Stochastic Fluidic Assembly of 3D Self-Reconfiguring Structures", Proc. Int. Conf. on Robotics and Automation (ICRA'10), Anchorage AK, May 2010, pp 2479-2484.

ture between the surface and the assembly substrate. By pumping liquid out through the assembly substrate and connecting the substrate's flow port to a portion of the structure surface, a flow sink is created, while pumping liquid into the tank through the substrate similarly creates a flow source.

Fully flexible flow routing through the interior of the already assembled structure requires one valve per connection surface in each module, that is six modules for a cube shaped module. This chapter is concerned with the development of a miniature fluidic valve that contains no moving parts, with the goal of implementing a small module for stochastic fluidic self-assembly. Chapter 6 of this thesis describes a proposed implementation of such a stochastic fluidic assembly system.

4.2 Thermorheological Fluids with Sol-Gel transition

Thermorheological fluids are liquids that change their flow properties depending on temperature. Here we look specifically at thermorheological liquids that undergo a reversible sol-gel transition, from low viscosity Newtonian liquid to gel, when heated above room temperature.

Poloxamers are a group of copolymers composed of a hydrophobic poly(propylene oxide) chains of length n with hydrophilic poly(ethylene oxide) ends of length n [153]. The notation $E_mP_nE_m$ is commonly used to represent a specific poloxamer. A wide selection of such copolymers exist, many of which are readily available commercially due to their frequent use as surfactants in industrial and medical applications. BASF markets poloxamers under the brand name Pluronic®. The thermorheological properties of poloxamers have been reported in numerous studies and show that when heated in aqueous solution above room temperature many exhibit a sol-gel transition [154–156]. The

relationship between viscosity (or elastic modulus for gels) and temperature is highly nonlinear and when heating continues after the gel-transition temperature is reached, most thermorheological poloxamer solutions eventually return to a liquid state. The exact mechanism of the sol-gel transition in poloxamers is not fully understood.

The fact that solutions of poloxamers in deionized water both have the described thermorheological properties and are non-conductive, makes them candidates for valving applications where the valve is a simple resistive heater. The fact that poloxamers have very low toxicity and are commonly used in medicines and dental care products additionally makes them easy to work with.

A first demonstration of fluidic assembly using the thermorheological effect of poloxamers already exists: Krishnan et al. apply the approach towards fluidic self-assembly described above, where flow is routed into or out of the surface of modules, at the microscale [157, 158]. In their demonstration a tile with side length 500 μm is placed in a flow chamber filled with a 15 % aqueous solution of Pluronic F127. An active seed tile contains three flow ports with a diameter of approximately 100 μm whose inlets can be heated through a resistive heater. When heated, the thermorheological fluid forms a gel inside the inlet port, effectively acting as a valve. This is enough to manipulate the flow through the chamber such that a free floating tile consecutively aligns different sides of the active seed tile.

The goal here is to scale the concept of a thermorheological valve to the millimeter scale to make it useful for autonomous cube-shaped modules in a three dimensional self-assembly system.

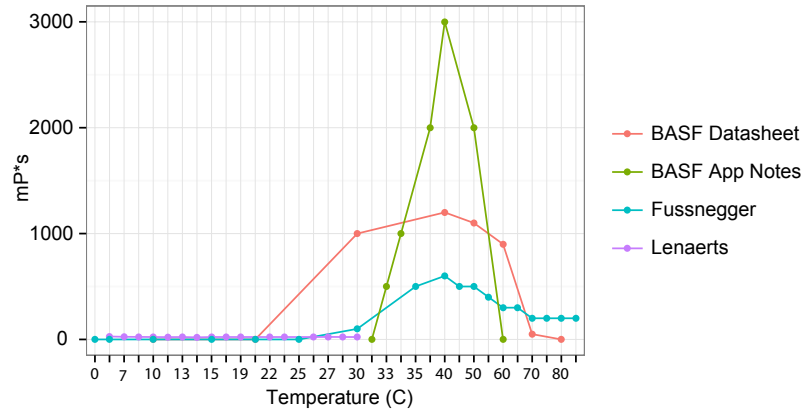
4.3 Fluid Characterization and Selection

The underlying principle of thermorheological valves is to "outsource" the mechanical switching function from the valve to the liquid: While in conventional valves a mechanical component of the valve, for example a plunger, blocks fluid flow in the valve closed position, flow through a thermorheological valve is blocked when the flowing liquid turns into a gel inside the valve. This implies that the valve performance depend largely on the liquid properties. To construct a valve that can operate with as high as possible applied pressures, one should select a liquid that transitions into an as hard as possible gel, and does so at a temperature tuned well to the temperature the valve is operating in.

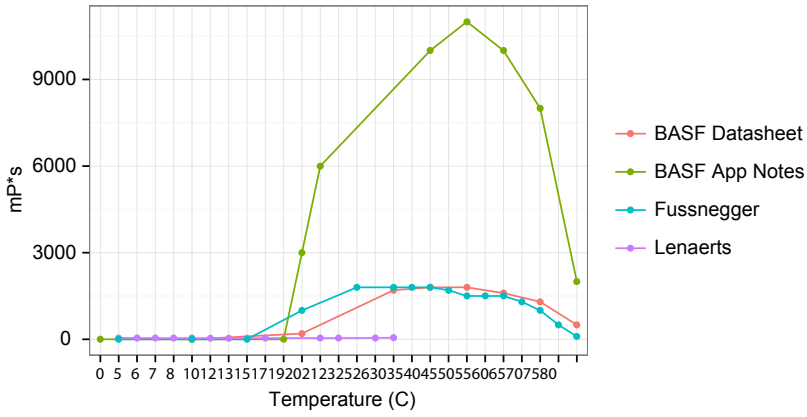
4.3.1 Quantitative Results from Literature

A number of candidate poloxamer solutions are suggested in literature for their strong thermorheological response that results in the liquids turning into gels of varying elasticity. Products named in this context are Pluronic F68, F127, and P123. Chaibundit et al. thermorheologically characterize Pluronic P123[1] and Pluronic F127 and show that certain mixtures form a hard gel when heated above temperatures between 20 °C and 40 °C [156]. Lenaerts et al. show that between 0 °C and 30 °C Pluronic F127 exhibits exponentially growing viscosity [154]. Additional information on the thermorheological properties of these products is available from vendor data sheets and application notes [159–161]. Figure 4.1 shows data for aqueous solutions of Pluronic F127 aggregated by visually estimating values from charts published in multiple sources. Note that Krishnan et al. use a 15 % Pluronic F127 solution for their aforementioned microscale assembly experiments [157].

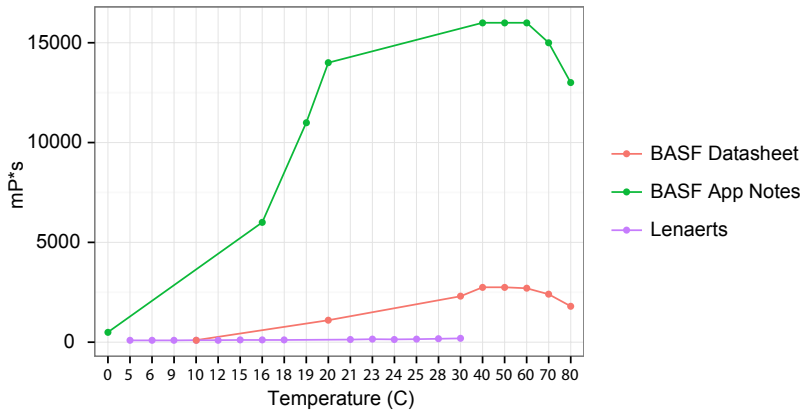
[1] For products under the Pluronic trade name, the first letter of the part number represents the phase of the product at room temperature. L = Liquid, P = Paste, F = Flake.



(a)



(b)



(c)

Figure 4.1: The thermorheological behavior of selected solutions of mixtures of Pluronic. Data aggregated from various studies and data sheets: Fussnegger [162], Lenaerts [154], BASF Datasheet [159], BASF Application Note [160]. (a) 15 %/16 % solution of Pluronic F127. (b) 20 % solution of Pluronic F127. (c) 25 % solution of Pluronic F127.

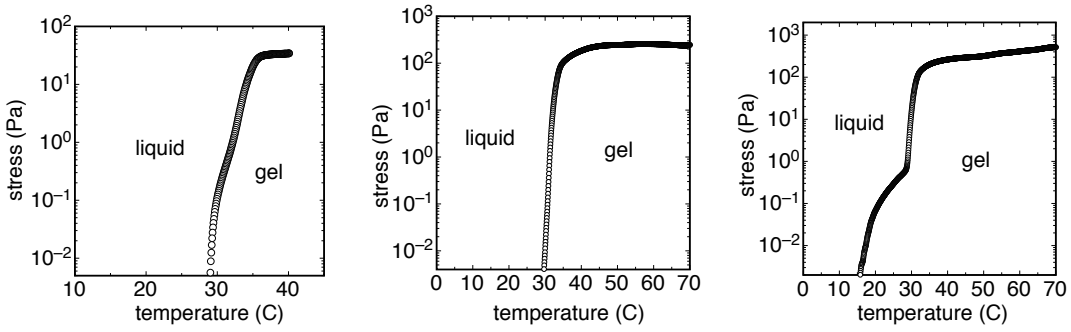


Figure 4.2: Phase diagrams for selected aqueous solution of Pluronic. (a) 15 % Pluronic F127, 85 % water. (b) 20 % Pluronic F127, 10 % Pluronic F68, 70 % water. (c) 20 % Pluronic F127, 15 % Pluronic F68, 65 % water. Figures prepared by Eric Brown, University of Chicago.

A valuable insight is provided by Fussnegger who summarizes research that shows that adding a Pluronic F68 component to F127 mixtures has strong effects on both the gel transition temperature and increases maximal shear modulus within the gel phase of Pluronic F127 [162]. According to the data presented there, unfortunately in the format of an imprecise magazine figure, shows that the viscosity increase of a mixture of 15 % Pluronic F127 and 20 % F68 ("15/20 mix") is approximately ten times as large as for 15 % Pluronic F127 only. However, the 15/20 mix also has a higher viscosity *before* the gel transition temperature is reached, which makes it too viscous to be stirred effectively in the assembly tank. A number of other mixtures of the same components are presented by Fussnegger, which are investigated further below.

4.3.2 Qualitative Mixture Selection

In the process of this project, several quantitative experiments were conducted by project collaborators at the University of Chicago in order to characterize Pluronic mixtures. Figure 4.2 shows selected preliminary results of these experiments. Ultimately it was

Table 4.1: Qualitative Observations of Properties of Various Pluronic Mixtures

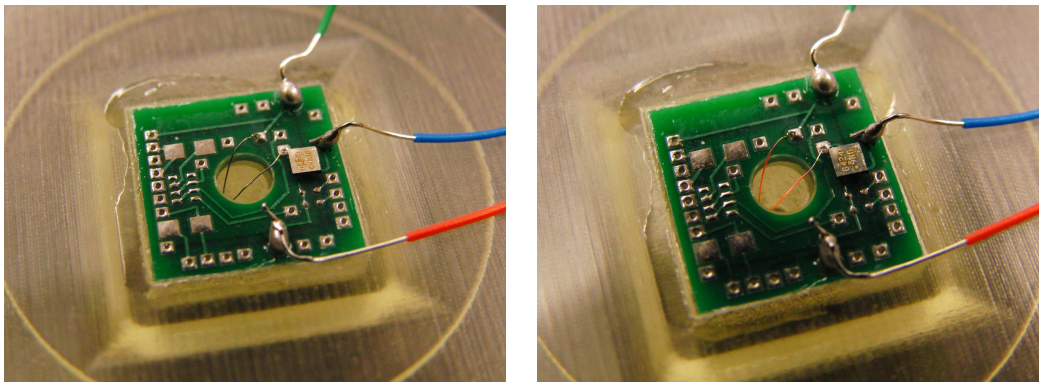
% weight F127	% weight F68	% weight DI water	observations
15	0	85	Mixes into transparent liquid[†] within half a day[‡]. Becomes a very soft gel at approx. 45 °C above which viscosity decreases.
15	5	80	Mixes into transparent liquid[†] within half a day[‡]. Shows some soft gel behavior around 55-60 °C above which viscosity decreases.
15	10	75	Same observations as for 15 %,5 % mixture
15	15	70	Mixes into transparent liquid1 within one day[‡]. Turns into a soft gel at 45 °C and into hard gel at 60 °C.
20	5	75	Mixes into transparent liquid1 within one day[‡]. Fast gel transition into hard gel at 40 °C.
20	10	70	Mixes into transparent liquid1 within one day[‡]. Fast gel transition into hard gel at 45 °C. Stiffness peaks at approx. 70 °C and decreases for higher temperatures while remaining in hard gel state.
20	15	65	Mixing took several days[‡], turns into clear liquid with small inclusions[†]. Flows viscously at room temperature, fast transition into hard gel at 40 °C.
20	20	60	Pluronic does not fully dissolve within one week. Very viscous soft gel at room temperature.

Qualitative observations on heating of various aqueous mixtures of Pluronic. "Soft gel" is a high viscosity fluid exhibiting flow upon manual stirring and shaking. "Hard gel" is a highly viscous gel that shows no flow upon manual mixing or stirring.
The heating is kept at approximately 5 °C per minute, hysteresis effects are not recorded.
[†] At room temperature.
[‡] While standing at approximately 5 °C.

decided to not pursue our own experiments on the characterization of the thermorheological effect. The data in Figure 4.2 are preliminary with high uncertainty and potentially erroneous and were only used to qualitatively inform the selection process.

Instead of the time consuming quantitative evaluation of mixtures that provides much more fine grained results than required for developing a thermorheological valve, additional mixtures were evaluated qualitatively with the results presented in Table 4.1.

Based on the information in literature, and the results summarized in Table 4.1, we selected a mixture of 20 % (by weight) of Pluronic F127, 10 % Pluronic F68, and 70 %



(a) (b)

Figure 4.3: Photographs of the single wire valve in its inactive and heated states.

DI water. We will refer to this solution as "Pluronic water" from here on. Pluronic F127 is $E_{101}P_{56}E_{101}$ using the notation previously introduced [160], Pluronic F68 is $E_{79}P_{28}E_{79}$ [161]. Both are available commercially in flake format for a price of US$2.15 per kg[2].

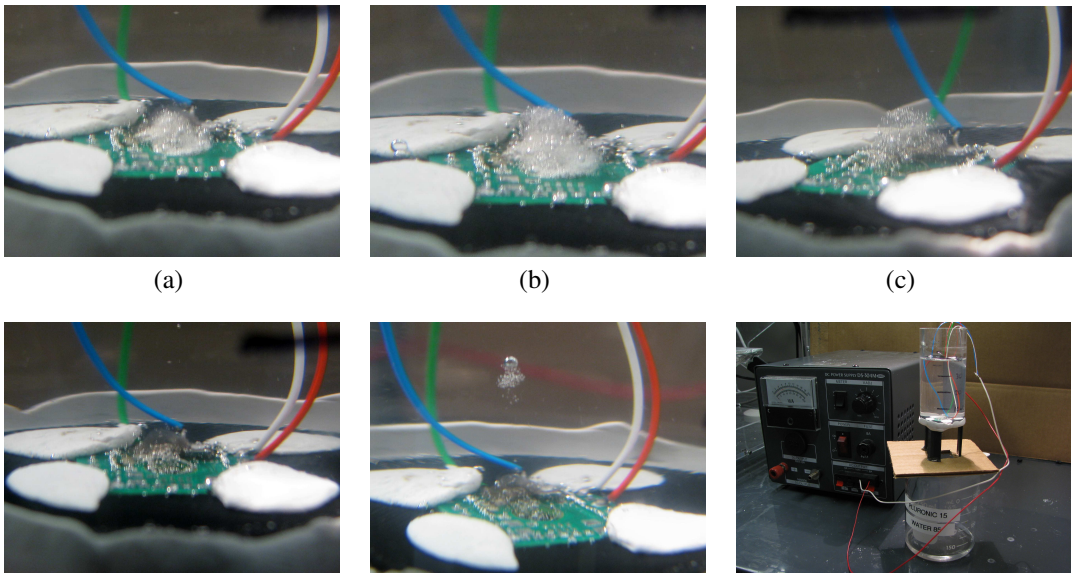
4.4 Implementations and Experiments

4.4.1 Simple PCB Valve

The first naive implementation of the mesoscale thermorheological valve is a printed circuit board (PCB) with a 2.8 mm diameter opening in the center and a 80/20 Nickel Chromium resistance wire for heating the flow through it as shown in off and heating states in Figure 4.3.

This simple valve implementation successfully stopped the flow of water out of a 40 mm high cylinder of Pluronic water as shown in Figure 4.4: The Pluronic water was

[2]Price quoted at time when experiments where performed in 2009.



(a) (b) (c)

(d) (e) (f)

Figure 4.4: Photographs of thermorheological valve during one heating and cooling cycle. Single PCB Valve under 40 mm H_2O pressure. (a) Initially white bubbling is visible as a small dome of gel is forming on the heated valve. (b) With time the gel dome with enclosed bubbles grows until a steady state is reached. (c) Once the heater is switched off, bubble formation stops and the gel becomes transparent. (d) As the gel cools the volume with enclosed bubbles shrinks. (e) In addition to dissolving, portions of the gel mechanically break off. (f) Overview of the experimental setup with a 40 mm high water column on top the thermorheological valve.

left to run through the 2.8 mm hole in the valve PCB, fed by gravity only. Then the valve was heated and after approximately 20 s the flow was stopped completely by a dome of gel forming on top of the valve.

This experiment allowed for closeup observation of the gel formation as shown in photographs in Figures 4.4(a) to 4.4(e). During heating, bubbles form around the heated resistive wire resulting in a dome shaped gel structure with enclosed bubbles. Once heating stops, the gel becomes transparent. As cooling begins, the gel structure dissolves and breaks up until the valve is cleared off the obstruction and flow resumes.

While this experiment successfully shows that a simple to implement thermorheological valve can in fact stop flow at the mesoscale, it needs to be noted that the pressure of 40 mm H_2O (400 Pa, 0.06 psi) is orders of magnitude below the pressures one would expect to encounter in a modular self-assembly structure with internal flow routing. Further, the flow rate in this experiment is far lower than what would be required to attract or repel modules to the assembly surface.

4.4.2 Valve Channel Geometry

The gel structure forming to block the simple PCB thermorheological valve in Section 4.4.1 above is only capable to sustain very low pressures. As soon as higher pressures are applied to the valve in closed state, the gel is quickly flushed out of the valve. If a higher pressure is applied to the open valve before heating is activated, no gel structure forms at the valve, most likely because any small volumes of gel that might form near the heating element are flushed out of the valve immediately.

Two approaches towards solving the problem of gel getting flushed out of the valve were explored: Methods to transfer more heat into the flowing Pluronic water to facilitate the formation of more gel, and modifications to the outlet of the valve where any gel that does form would accumulate and ultimately block the overall flow.

The experiment performed involves applying a series of fixed pressures to the inlet chamber and collecting the flow through the valve with the valve off and with the valve heated, respectively, in separate experiments. The differential pressure between inlet and outlet chamber was varied between runs of the experiment and the time taken to collect a predefined amount of pluronic water downstream of the valve was measured for each run. A visualization of the experimental setup is shown in Figure 4.5. Good

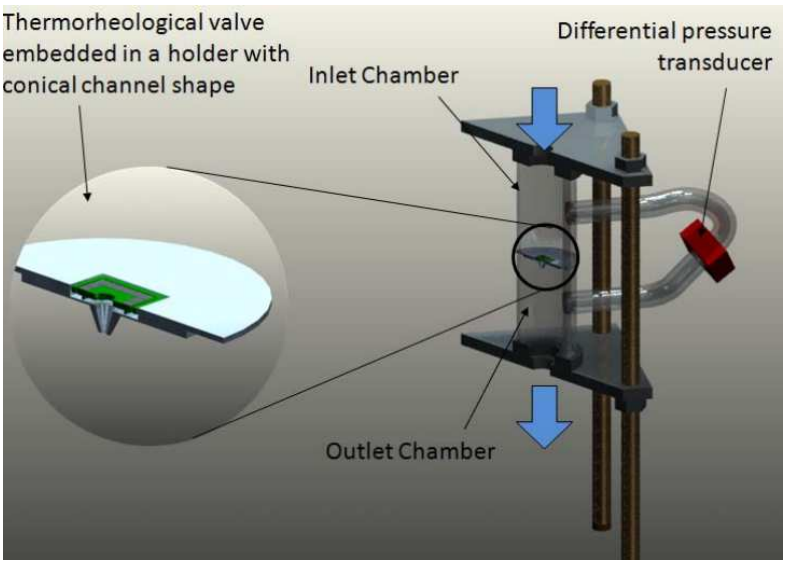


Figure 4.5: Schematic of experimental setup for valve geometries evaluation. The thermorheological valve is placed in the middle of a cylindrical transparent acrylic pipe. The differential pressure between the two chambers is measured by a pressure transducer connected to both halves of the chamber. Pressure is applied at the top, the bottom end of the pipe dispenses into a container whose weight is recorded to determine the flow rate.

valve performance is indicated by a significant reduction in flow rate at constant pressure when the valve is switched on compared to the case of undisturbed flow with the valve off.

The various valve geometries for which tests were performed are shown in Figure 4.6 as 3D rendering and drawing respectively. Figures 4.6(a)-4.6(d) are designs that use the simple PCB valve described above. For the other channel designs a resistive temperature detector (RTD) is used as heating element. RTDs are temperature sensors and therefore designed for efficient heat transfer between the resistive sensor and surface. This, and the fact that they have a suitable resistance, makes them effective, but expensive, resistive heaters. Figure 4.6(k) is a photograph of the RTD used in this experiments. Finally, the valve geometry in Figure 4.6(i) uses an RTD heating element but extends the heater surface by inserting a copper pipe into the valve and using heat conductive silver paste

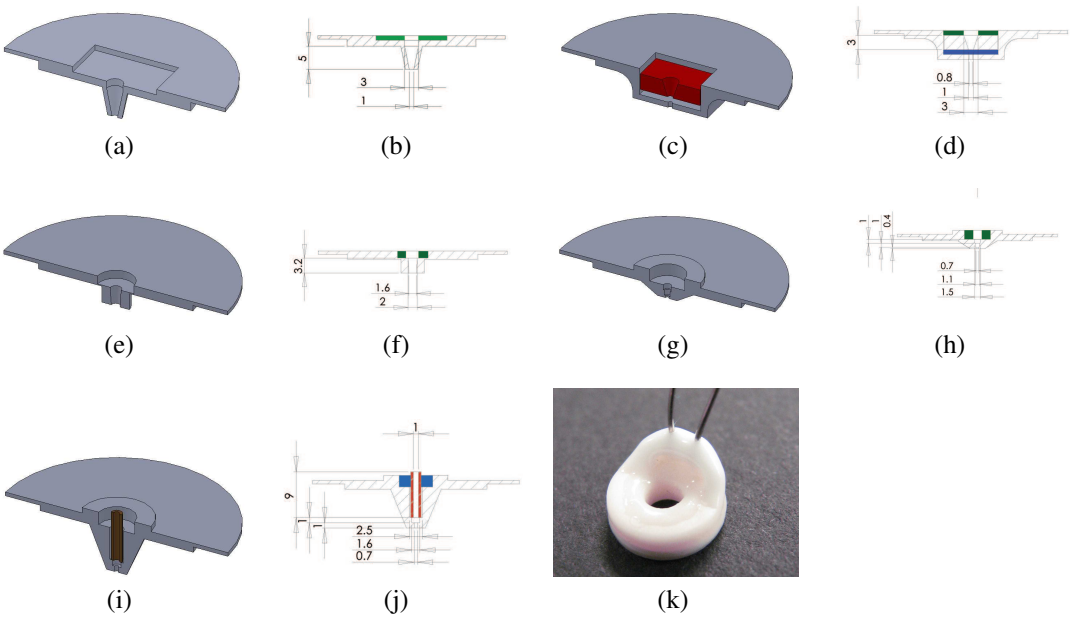


Figure 4.6: Different valve geometries for thermorheological valves used in the setup shown in Figure 4.5.

for forming a good thermal connection.

For the conical design in Figure 4.6(a) no significant difference in flow rate was measured between undisturbed flow and flow through a heating valve. This appies even when a 10 s period of pre-heating with externally stopped flow is added. Plots with measurements of all runs with undisturbed flow (marked as blue square data points) and flow through the heated valve (marked as red triangle data points) are provided in Figure 4.7(a) for the experiment without preheating and in Figure 4.7(b) for the experiment with preheating.

The design in Figure 4.6(c) contains a filter cloth separated from the simple PCB heater implementation by a spacer (red in Figure 4.6(c)). Despite testing several types of filter material ranging from paper coffee filters to various types of cloth, no mesh could be found that lets room temperature Pluronic water pass for longer than a few seconds before blocking.

All experiments with RTD heaters were only performed with a preheating step during which no pressure was applied. For the design with a straight channel shown in Figure 4.6(e) no significant difference in flow rates between the undisturbed flow and the flow through a heating valve were observed. Figure 4.7(c) shows all measured data points for this experiment.

More promising results are achieved with the final two RTD heater designs: For the narrow channel design in Figure 4.6(g) a small difference in flow rate is detected for small applied pressures of around 0.4 psi. While the undisturbed flow of 50 ml through the valve has a duration of 320 s, the flow with the valve on takes 430 s at this pressure. For the design with an extended heater surface in Figure 4.6(i) a similar effect is observed at pressures up to 1.0 psi. The measurement plots of the final two experiments are shown in Figures 4.7(d) and 4.7(e), respectively.

In conclusion, the mesoscale thermorheological valve fails to interrupt flow for applied pressure above 2 psi. However, it appears that design changes that facilitate heat transfer into the Pluronic water and a narrowing channel downstream of the heater improve the efficiency of the valve.

4.4.3 Flow Switching

After performing the previous experiments with PCBs to which a resistive wire heater is manually soldered, the following experiments are performed with an overall similar PCB where the heating element are four surface mount resistors. Here, the valve PCB consists of four 1Ω resistors in parallel to achieve a 0.25Ω resistance across the resistor array. These resistors are mounted on PCB with a thickness of 1.6 mm. The PCB dimensions are 14 mm by 14 mm with a hole of diameter 2.8 mm in the center. As

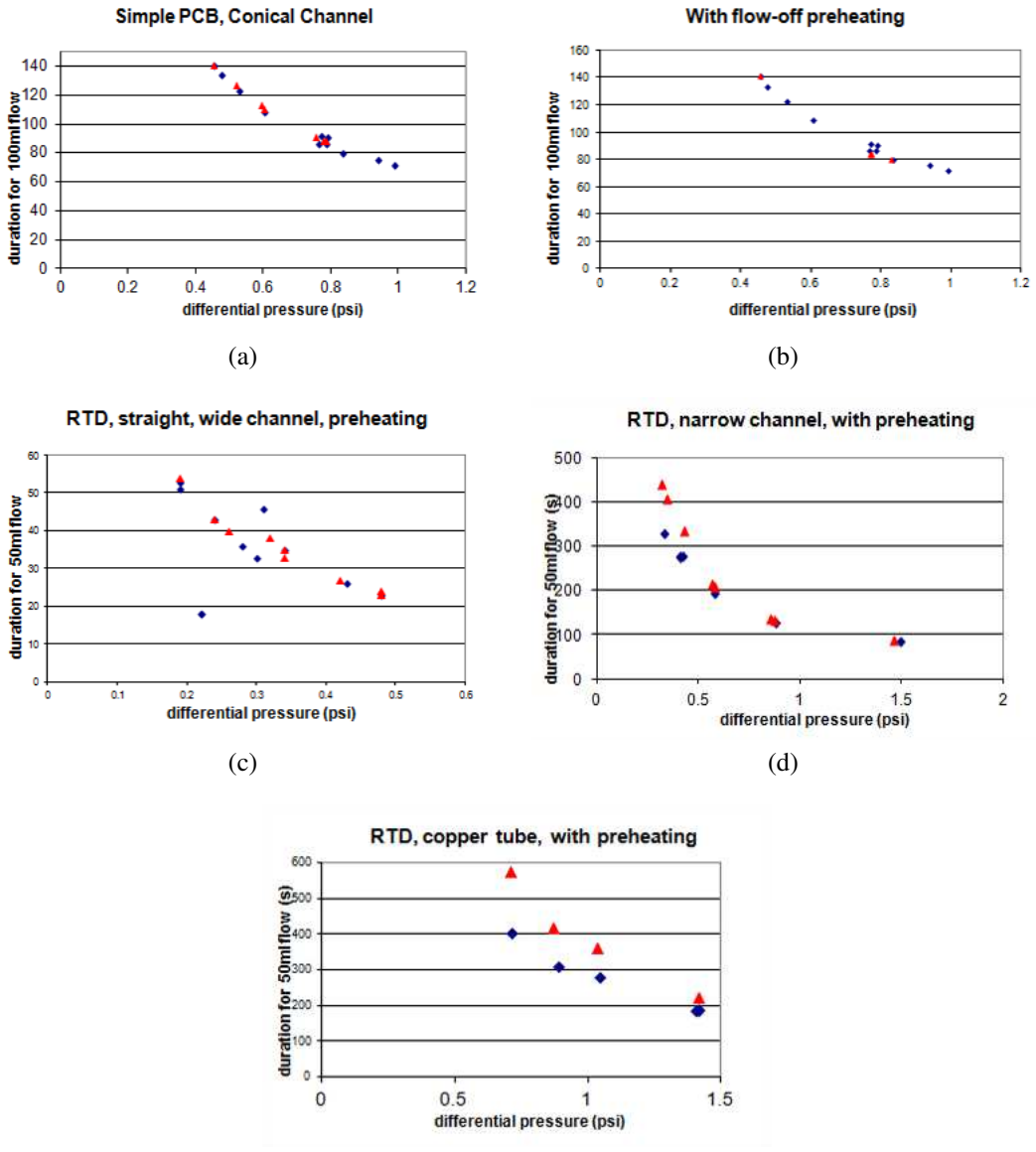


Figure 4.7: Fluid flow through selected channel geometries and heater types. The duration it takes 50ml of Pluronic water to flow through a valve are plotted with respect to the applied differential pressure across the thermorheological valve. Blue square data points represent measurement runs with undisturbed flow, i.e. no heating. Red triangle data points represent experiment runs through the active valve, i.e. with heaters on.

before, each face is supplied with 2 V DC supply limited to 2 A. The resistors are exposed directly to Pluronic water.

Here we use insights from the valve channel experiments above to switch the flow of Pluronic water with an applied pressure off. The experimental setup is identical to that in the previous Section 4.4.2 with the flow rate determined by computing the numerical derivative of scale readings taken at approximately 1 Hz.

Figure 4.8 shows results of three separate experiments. After 60s of continuous flow, the valve was heated (switched to closed). A single valve only significantly reduces flow up to 4.8 kPa. An arrangement with a fine metal grid mesh placed across the flow to facilitate heat conduction into the fluid, however, stops flow for pressure differentials up to 9.0 kPa at the expense of reduced flow in the valve open position. Two valves in an array with no mesh stop flow at pressure differentials up to 6.2 kPa but significantly reduce flows up to 9.0 kPa (1.3 psi). The latter two options are both feasible for an application in a self-assembly system and the selection depends on the more frequent valve position during assembly. If an assembly strategy primarily keeps valves closed, the option using metal meshes would be preferred, for strategies with the valves mostly open, the double-PCB option would be preferred.

4.4.4 Flow Routing

Experiments so far have focussed on switching flow on or off with the use of the thermorheological valve. For the intended application scenario inside a fluidic self-assembling structure, only flow routing is required. This means that reducing an applied flow to zero and withstanding the applied pressure would never be required from the thermorheological valve. A sufficient performance would be to withstand the pressure

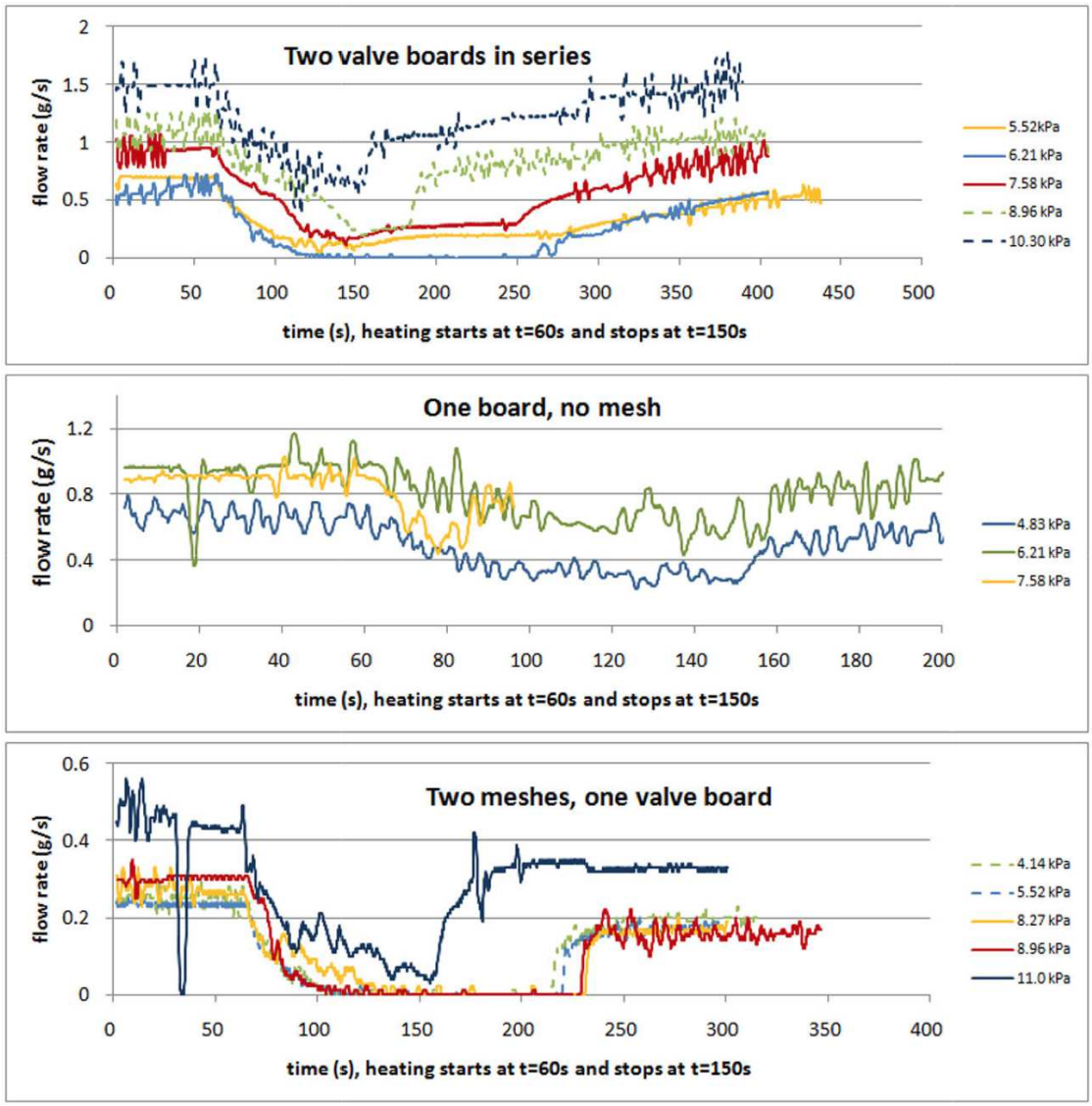


Figure 4.8: Evaluation of thermorheological valve with mixture of 20 % (weight) Pluronic F127 and 10 % F68. Current through the heating resistors switches on after 60s, is then pulsed to keep the liquid temperature in vicinity of the valve at approximately 65 °C to 75 °C. With one valve circuit board and two meshes across the flow pressures up to 9.0 kPa could be stopped, for 11.0 kPa the flow rate could be reduced significantly. For a single valve board with no mesh the flow in the open position is much higher but the valving performance is degraded. Two valve boards in series with no mesh across the flow have similarly high flow in the open position but can stop flow at pressure differentials of up to 6.2 kPa and lead to a significant reduction of flow rate at all other pressures measured.

differential between the blocked outlet and an alternative outlet the flow is routed to.

The following experiment applies this insights to demonstrate the flow switching capabilities of the mesoscale thermorheological valve using the simple PCB design described above in Section 4.4.1. Figure 4.9 represents a time sequence of photographs throughout the experiment: Initially, Pluronic water flows through both outlets of the splitter device with an applied pressure of approximately 2 psi (Figure 4.9(a)). The thermorheological valve in the left flow outlet is now activated (Figure 4.9(b)) and the flow of Pluronic water is blocked upstrem, external to the system shown (Figure 4.9(c)). As pressure is reapplied by unblocking the flow upstream only the right outlet lets Pluronic water pass while the left outlet is blocked due to the valve activation (Figure 4.9(d)). Approximately 10 s after the left valve is switched off and heating stops, soft gel is pushed through the left valve (Figure 4.9(e)) and another 5 s later flow resumes through the left valve (Figure 4.9(f)). Figures 4.9(g) to 4.9(k) show the same sequence applied to the valve on the opposite site of the setup.

This experiment successfully demonstrates that flow can be switched between multiple outlets for applied pressure of 2 psi if preheating with no applied flow is performed.

4.5 Discussion

The goal of implementing a thermorheological valve at the mesoscale, that is for flow diameters in the millimeter range, has been partially achieved. The series of valve designs presented in this chapter have the capability of redirecting flow between different outlets, stopping flow at flow rates up to 1.0 g s^{-1} and applied pressure up to 6.2 kPa, and reducing flow at pressures up to 9.0 kPa (1.3 psi). A key step towards achieving these results was the selection of a suitable mixture of poloxamers that forms harder gels than

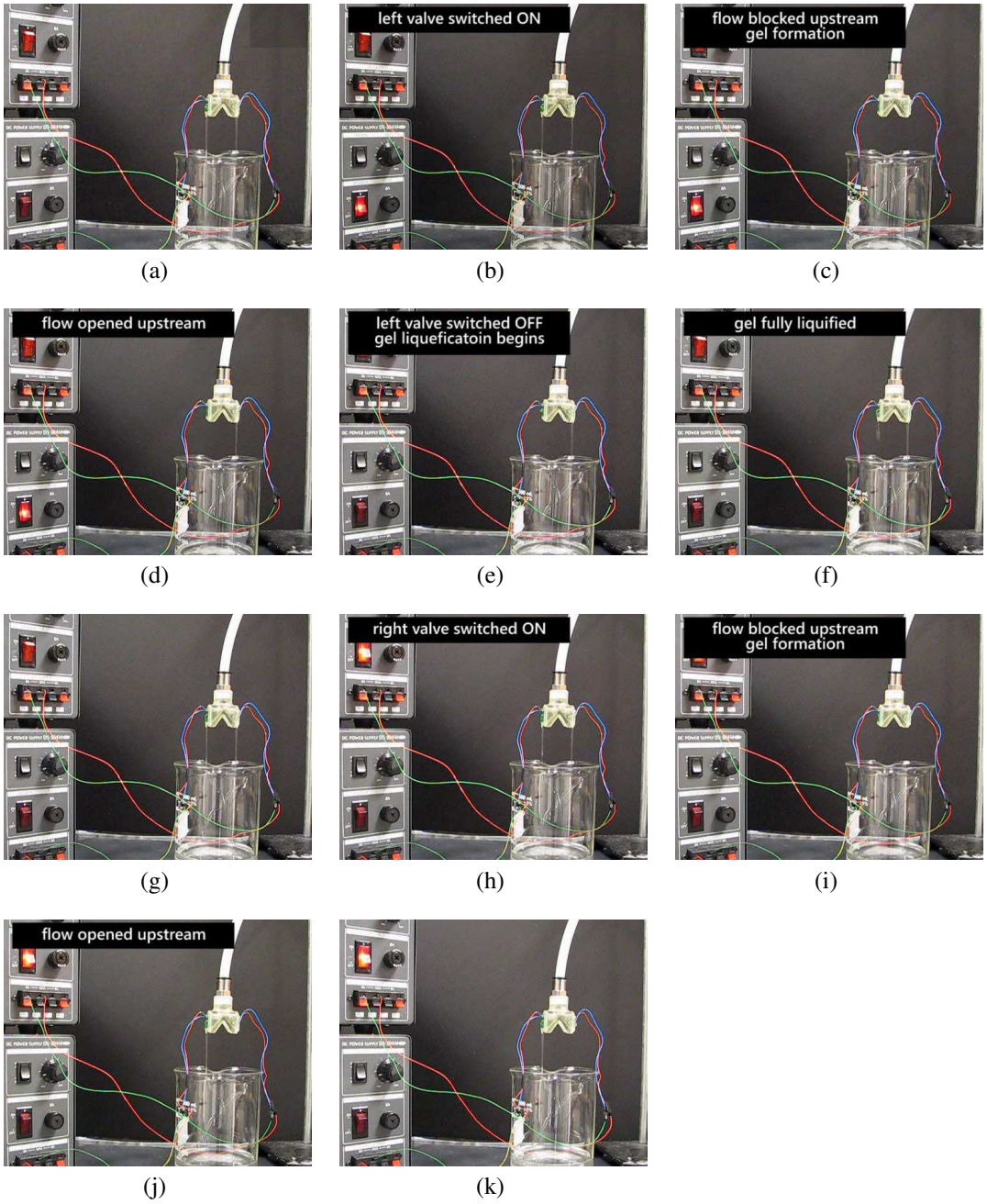


Figure 4.9: Sequence of photographs of "flow routing" experiment. (a) Pluronic water flows into a flow splitter with two outlets from the top of the image. (b) The left valve is activated, and (c) all flow is externally stopped. (d) After re-applying flow, the left outlet is blocked. (e) The left valve is switched off and (f) flow resumes through the left outlet. (g)-(k) Repeat of the same sequence of steps with the right outlet.

those previously used for microscale self-assembly experiments, to which it might also be applicable.

It remains unclear whether the thermorheological valve's capability meet the requirements of flow routing in a 3D stochastic fluidic assembly demonstration. Chapter 6 will discuss the thermorheological valve in the context of this application.

CHAPTER 5

COMMUNICATION IN SELF-RECONFIGURING MODULAR ROBOTS

5.1 Communication as a Prerequisite for Self-assembly

Communication between modules is a requirement for any self-assembling system. Only with information about the goal and current state of the assembly process can modules participate actively in the assembly process. Assumptions about the amount of information required vary: Only using information about the presence of neighbors (a concept known as stigmergy) has been shown sufficient to create some biologically inspired structures [163]. On the other extreme are self-assembling structures where each module has a complete state representation of the entire assembly such as the *MTRAN II* system [83]. This chapter aims to provide an overview over the design decisions involved in creating a communication architecture for a self-reconfiguring system. The first part of the chapter provides a review of communication architectures reported in literature. Subsequently, a specific implementation of communication suitable for a system with small non-autonomous modules, like those in Chapter 6, is presented.

5.2 Communication Architectures

Communication architectures for modular robots can be classified with regard to their transmission technology, the network topology, and the type of communication.

5.2.1 Transmission Technology

The most basic design decision for a communication architecture is the choice of transmission medium, usually implying the transmission technology. Examples for transmission technologies are wired communication using differential voltages, optical communication with infrared (IR), and radio frequency (RF) data transmission. In Table 5.1 *electrical* refers to systems using electrical contacts to transfer data between modules, whereas *wired* is used for system where pluggable wires are used for communication. *IR* refers to infrared optical communication while *optical* refers to systems using visible light.

5.2.2 Network Topology

Network topology refers to how the network is connected and therefore defines which modules of a self-reconfiguring structure can communicate with each other. There exist, for example, star topologies where each module is only connected to a central controller, bus topologies where all clients are connected to one set of conductors carrying data, and mesh topologies where some clients are directly connected to each other but any transmission route might include clients acting as relays. The choice of transmission topology has implications on the topology: RF communication usually broadcasts information (unless the transmitter is directed) and is therefore equivalent to a bus topology where every node receives all data. Any optical communication in a changing structure is restricted to communication between neighboring modules since the required line of sight connection to other modules could be obstructed by other parts of the structure.

5.2.3 Communication Schemes

A final classification for communication architecture, the type of communication, is defined on the software level by the routing protocol. This is the logical topology of the network (in contrast to the physical topology mentioned above). The following four communication schemes are used in the literature on modular robotic systems:

- Global communication where each module can and does communicate with arbitrary other modules.
- Master-slave is a type of global communication where only communication between one master and any module is permitted. This is the logical equivalent to the physical star topology.
- Hormone inspired communication as suggested by [4] is also initiated by a central master but data is passed from module to module spreading throughout the entire structure instead of being targeted at one destination and transmitted directly.
- Local communication is where modules can only communicate with modules within a certain range of their own position, for example only their direct neighbors.

The choice of physical network topology does not pre-determine the type of communication: A bus topology, for example, allows for local communication by restricting access to messages in software. Conversely, a mesh topology allows for global communication through forwarding messages over multiple modules.

5.2.4 Standard Protocol Stacks

To ensure interoperability between devices, many standards exist that define a stack of technology, topology and communication type. For example, the proprietary CAN bus standard–used by some modular robots–defines the use of an electrical connection, the voltage level of the signal, the message protocol, and the behavior of each communication node. Where a system uses a well defined standard, the standard is listed in the *technology* column of Table 5.1. In some cases, imprecise naming conventions for standards have become generally accepted. While technically IrDA stands for the *Infrared Data Association*, a standards body, the acronym is conventionally used to refer to any subset of the standardized protocol stack released by the IrDA. In the following review, it is assumed that systems reported to use IrDA only follow the the physical layer specification. Similarly, Bluetooth refers to a set of standardized protocol stacks; unless further information is available, we assume only the physical layers are used.

5.3 Review of Communication in Self-reconfiguring Robots

Table 5.1 lists selected self-reconfiguring modular robotic systems and the types of communication architecture for each system. Only systems with a reported physically implemented communication architecture are included. Relevant literature for each system is cited in the table; further sources employed in compiling the table are the papers by Yim et al., Ostergaard et al., and the book by Stoy et al. [5,11,87]. The following section provides a review of design decisions made for existing systems.

Table 5.1: Overview of Comm. Architecture for Selected Modular Robots

System	Affiliation	Technology	Topology	Type of Comm.	Power Supply	Source
ATRON	U.S. Denmark	IrDA†	Mesh	not published	on module, shared	[86]
Catom	CMU	IrDA, 802.15.4 (Xbee)	Mesh, Broadcast	master slave, local	distributed (inductive)	[164]
CEBOT	Nagoya	Electrical, IrDA	Parallel Bus	master slave	on module	[165]
Chobie II	TiTech	Optical	Mesh	global	on module	[85], [84]
CKBot	UPenn	Electrical, IrDA	CAN bus, Mesh	global	distributed	[166]
CONRO	USC/ISI	Electrical, IrDA	Mesh	hormones	on module	[67], [68]
Fracta	MEL	IrDA	Mesh	local only	external	[2]
I-Cube	CMU	Wired	RS232 Bus	master slave	on module	[134]
Miche	MIT	IrDA	Mesh	local & master slave	on module	[24]
Micro-Unit	AIST	Wired	Star	master slave	external	[133]
Molecube	Cornell	Electrical	RS-232 Bus	global	distributed	[21], [167]
Molecule	Dartmouth	Electrical	RS-485 Star	not published	distributed	[76]
MTRAN	AIST	Electrical	RS-485 Bus	master slave	distributed	[81]
MTRAN II	AIST	Electrical, RF	CAN (RS-485 Bus)	global	on module	[83]
MTRAN III	AIST	Bluetooth	CAN	various, address based	on module	[109]
Odin	U.S. Denmark	Bluetooth	CAN	various, address based	on module	[109]
Polypod	Stanford	Electrical	RS-485 Bus	master slave	distributed	[60], [38]
PolyBot	PARC	Electrical	CAN Bus‡	various	on module	[3], [4]
Prog. parts	U. Wash.	IrDA	Mesh	local only	on module	[107]
Replicator	EU	Electrical	CAN Bus	global	on module	[168]
Stoch. Tiles	Cornell	Electrical	Serial bus	N/A	distributed	[22]
Stoch. Cubes	Cornell	Electrical	Dallas 1-Wire bus	local only	distributed	[23], [169]
Superbot	USC/ISI	IrDA	RS-232 Mesh	local only	on module, shared	[170]
TeleCube	PARC	IrDA	Mesh	not published	distributed	[104]
YaMoR	EPFL	Bluetooth	Mesh (RF)	master slave	on module	[147]

† ATRON implements IrPHY and higher layers of the IrDA stack [86]. ‡ PolyBot uses a semi-global CAN bus [3].

5.3.1 Choice of Topology

Literature indicates that mesh topologies with neighbor communication only are preferred over other physical and logical topologies, primarily due to their favorable scalability. If only neighbors exchange information, the structure can be expanded infinitely without any bandwidth bottlenecks occurring. In surveying implemented systems, however, it becomes quickly apparent that topologies where each module can directly communicate with a central entity are preferred because they simplify experimentation with the system. *Catoms* [164] and *CKBot* [166] choose the benefits of both by using a mesh topology for regular communication and mesh and broadcast topologies respectively for maintenance tasks. *Odin* is unconventional in that it can change its network topology on the fly [171].

5.3.2 Wired and Optical Technologies

Electrical contacts and IR are by far the two most common transmission technologies used in self-reconfiguring robots, probably due to their low energy consumption and small footprint. Both place constraints on the connector design: Electrical connections require a good mechanical connection between modules, whereas IR transmission only requires space for transmitters and receivers on adjacent surfaces of communicating cubes. IR can also be used for non-connected modules as long as the receiver is within the transmission cone of the transmitter. *CKBot* has two separate modes for communication between connected modules and non-connected ones up to a distance of 2m [166]. *CONRO* modules extend the use of IR even more by using it for communication in connected and separated states as well as distance sensors [68].

5.3.3 RF Technologies

RF technologies are often favored over wired ones because they loosen requirements on mechanical connector design, but they do come at the price of higher power consumption. Three more recent systems use RF: *MTRAN III* and *YaMoR* use Bluetooth while *Catoms* is the only fully developed modular robot system that uses an IEEE 802.15.4 based technology geared towards wireless mesh networking. Bluetooth is not inherently a candidate for systems made of many modules: Due to its 3-bit address space, a Bluetooth picocell can only communicate with eight active nodes (and host 255 dormant nodes). This makes the use of a mesh topology necessary (referred to as *scatternet*) for which no generally agreed upon standard exists. *YaMoR* implements such a Bluetooth scatternet while MTRAN III's use of Bluetooth is not reported.

Kuo and Fitch argue in [172] that standard mesh networks are not well suited for large scale modular robot systems because they do not allow several nodes to transmit in parallel and their communication capacity quickly saturates. Instead, they propose a single channel wireless solution with constant bandwidth and neighbor-to-neighbor communication only [173].

5.3.4 Other Considerations

Several systems implement multiple communication architectures. CEBOT's modules are autonomous before connecting and communicate optically before assembling and through electrical connectors on a bus once attached [4, 60, 165]. *Catoms* use two sets of communication architecture for global (maintenance) and local communication. *MTRAN II* modules contain unspecified RF receivers for receiving broadcast commands from a central controller as well as electrical connectors for inter-module communica-

tion on a bus [83]. Further, both *MTRAN II* and *MTRAN III* feature two sets of electrical connectors, one for global communication on a bus and one for local communication between neighbors only. *MTRAN III* modules contain hardware for Bluetooth, CAN, and IR communications in each module, but only the CAN bus is reported as used in [109]. Similarly, Superbot also includes an RF link for control through a handheld computer.

5.4 Implementation for Small Non-Autonomous Modules

The approach to robotic self-assembly and reconfiguration here is driven by the goal to miniaturize and simplify the individual modules. The term *programmable matter* introduced to the context of robotics by Goldstein et al [9] illustrates our notion that, ideally, self-reconfiguring systems are made of a large group of homogeneous modules, each being as simple as possible. A module that can be produced cheaply and distributed in bulk is the ultimate goal of this endeavor. Specifically, the communication system described here has to be suitable for a fluidic stochastic assembly system where a large number of modules floats passively in an assembly medium before being attracted and connected to a self-assembling structure. The size of each module should be below an inch across on its widest side.

5.4.1 Single Wire Serial Bus

Because an electrical connection between neighboring modules is required for power distribution already, the use of electrical connectors for communication suggests itself. The optical and wireless alternatives would furthermore add additional complexity to each module because they require transmitters and receivers to operate.

The connection of each module face's communication pins inside the module defines the network topology. If the faces are not connected internally, a mesh topology results, if all faces are connected a bus topology results. The notion that a bus topology unreasonably limits scalability of the system is far fetched even for systems with a large module count: While assembly in a typical existing self-assembling system events occur on the order of seconds, messages are transmitted at time scales of milliseconds or less. Even if every assembly event required transmission of a message to every module, a bus topology would scale to $O(10^3)$ modules, far exceeding the currently accomplished numbers of connected modules for all reported systems. With a more efficient message protocol, this limit would be extended by further orders of magnitude.

Contrary to the theoretical disadvantage of poor scalability for bus topologies, a mesh topology comes with the disadvantage of requiring six times more serial ports or a multiplexer per module adding to hardware and software complexity. Because complexity of modules is a more prevalent concern than reaching module counts exceeding thousands of modules a bus topology is determined as most suitable for a module with minimal complexity.

5.4.2 Communication Protocol

The bus topology allows every module to communicate with any other module in the structure as well as the central controller in the substrate. This allows for the implementation of communication protocols ranging from centrally controlled master-slave protocols to local communication where only neighbors exchange data, while retaining a constant speed of transmission. For an initial demonstration of self-assembly, a protocol that predominantly relies on an external central controller moderating all infor-

Table 5.2: Message Structure

Byte	Description	Content
0-4	Message Header	0x07 0xFF 0xFF 0xFF 0x0E
5	Target Address	0x00 to 0xFF
6	Origin Address	0x00 to 0xFF
7-8	Command	see Table 5.3
9-13	Data	see Table 5.3
14-18	Message Tail	0x07 0xFF 0xFF 0xFF 0x10
19	Checksum	XOR of bytes 0-18

Table 5.3: Message Types

Command	Description	Data Bytes
CS	Cube Status	5 bytes for heater, valve and neighbor status
PG	Ping/Request Cube Status	0x00
SH	Switch Heaters On/Off	0x00 to 0x40 for heater status
SV	Switch Valves Open/Closed	0x00 to 0x40 for valve status
SL	Switch Status LED On/Off	0x00
IO	Initiate Orientation Detection	0x00
DO	Detect Orientation of Neighbor	0x01 to 0x06 defining which neighbor
OR	Orientation of Neighbor	0x01 to 0x18 defining orientation

mation flow was chosen. The communication with newly acquired modules is the only exception where the special local communication sequence described below is used.

The message structure of the protocol for global communication in our system is shown in Table 5.2 and a list of currently implemented message types is given in Table 5.3. When using a communications bus, conflict resolution, message handling, and module identification are critically important. Each module receives every message which includes a predefined header, footer and a checksum. For each message every module confirms the presence of the header and footer, and validates the checksum. After malformed data is discarded the module compares the message's target with it's own unique address and processes the message if they match.

Upon connecting to the assembly and powering-up the module is initialized with a default address of 0xFF. Before a new module receives a *CA* command, no second new

module can be connected to the structure as the two new modules would be undistinguishable. The address 0x21 is reserved for broadcast commands. To eliminate conflicts, only the central controller is permitted to initialize communication. The central controller can send any message type listed in 5.3 except *CS* and *OR*. The target module will change it's state if instructed and always reply with a *CS* message describing the current state of the module.

5.4.3 Orientation Detection

In a stochastic fluidic assembly system, a module only receives power, initializes its controller, and is assigned an address after connecting to the self-assembling structure. Once connected to the self-assembling structure, it is each module's task to selectively attract further modules to locations on the current structure surface in order to continue assembly of a target shape. However, the newly acquired module does not inherently have knowledge of its orientation relative to the other members of the structure. The detection of relative orientation between two cube shaped modules can be broken down into two questions:

- Which faces of the two cubes are facing each other?
- What is the relative rotation about the axis perpendicular to the plane in which the two cubes connect?

Figure 5.1 explains how arranging the four separate communication pins in different non-cyclic permutations on each face, results in a unique pairing of pin numbers between fixed and newly attached modules. Prerequisite for this scheme is that the communication bus is connected to four microcontroller I/O pins, which normally operate

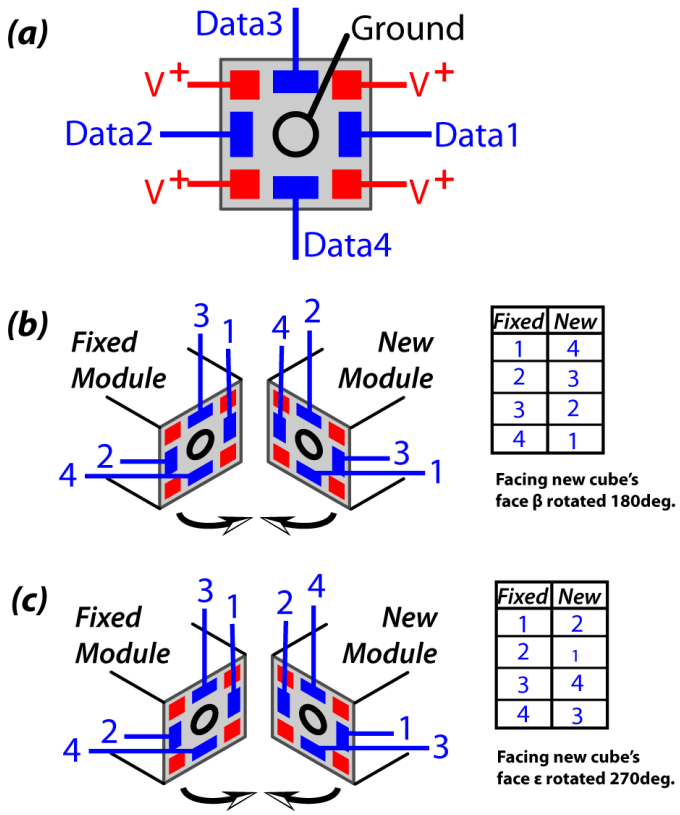


Figure 5.1: Schematic of a rotationally symmetric connector and the protocol used for detecting the relative orientation of two connected modules. Every face of the cube shaped module has four pins for data transmission. (a) To be able to detect the relative orientation of neighboring modules, the order of data pins on each face are non-cyclic permutations of each other. (b) When two modules are facing each other, the matched pairs of pin numbers are unique for each relative orientation between the two cubes. (c) Rotations about the axis orthogonal to the connecting module faces result in cyclic permutations of the paired pins.

in tandem, but switch to separate functions during the orientation detection sequence. The pairs are detected by independently switching each data pin on the newly attached module to a high voltage in ascending order, while all other pins are kept at ground respectively. The neighboring module that is already part of the structure and whose orientation relative to the base substrate is known, detects in which order its own data pins are at the high voltage. The resulting four ordered pairs of pins numbers on two neighboring modules are converted to orientation information using a lookup table. The

orientation detection routine is triggered by the base controller sending the commands *IO* and *DO* to the two involved modules. Both messages are acknowledged with a *CS* messages followed by the orientation detection whose result is reported back to the base controller with a *OR* message. The numbering of pins is by arbitrary convention.

The drawback of this orientation detection scheme is that due to operating on a communication bus, message can "loop back" to the sender if a loop exist in the assembled structure. This would be the case whenever the newly attracted module whose orientation is to be detected, has two initial neighbors.

5.4.4 Module Electrical Design

All modules are electronically equivalent and only differ in the address stored in volatile memory once connected to the structure. At the center of the electronic configuration is an Atmega48 microprocessor. The microprocessor uses its built-in UART to communicate with the other cubes via the serial protocol described in the previous section. To enable the automatic orientation detection routine described above, the four redundant signal lines are connected to different pins of the the microcontroller. Only one line is connected to the receiving UART pin and used for receiving messages. The remaining three are connected to digital input pins and used exclusively for orientation detection. Since we are using a single wire bus architecture but UART is designed to have separate lines for transmit and receive, both UART pins are connected together. To avoid a module "hearing" itself talk, a diode network is utilized to properly direct transmitted and received messages to the appropriate UART pins on each cube's microcontroller. The communication subsystem operates at a voltage of 1.8 V.

5.4.5 Base Controller Electrical Design

The base controller capable of sending messages to any cube in the assembled structure through the assembly tank substrate is PC software written in C#. An RS-232 level converter between the PC and the self-reconfiguring structure is required: While the serial port of a PC operates at up to 25 V as specified by RS-232, the communications bus inside the modules operates at up to 1.8 V. Because transmission and receiver lines on each cube are not static, a single communication line from the voltage level converter connects to all four of the lines emanating from a single cube. Using a diode network, it is possible to send and receive signals from the base station using a single communication wire. This configuration results in a local echo on the computer software but no echo on the modules, the former being easily removed in software.

5.5 Experimental Validation

To validate the functionality of the communication architecture described above, the electrical components of five modules were connected in three arrangements as shown in Figure 5.3. While the linear arrangement served as an initial experiment to evaluate the consistency of voltage levels when modules are added, the loop and mesh arrangement serve to simulate arrangements likely to be encountered in a real self-assembly scenario. As shown in Figure 5.2, the test was performed with only the electronic components of modules, including the flexible PCB of the main controller and connector PCBs, which are plugged into the main controller PCB. These are the same components as those revisited later in this thesis in Chapter 6 where the complete module design is discussed.

A first indication of the soundness of the electrical design of the communication ar-

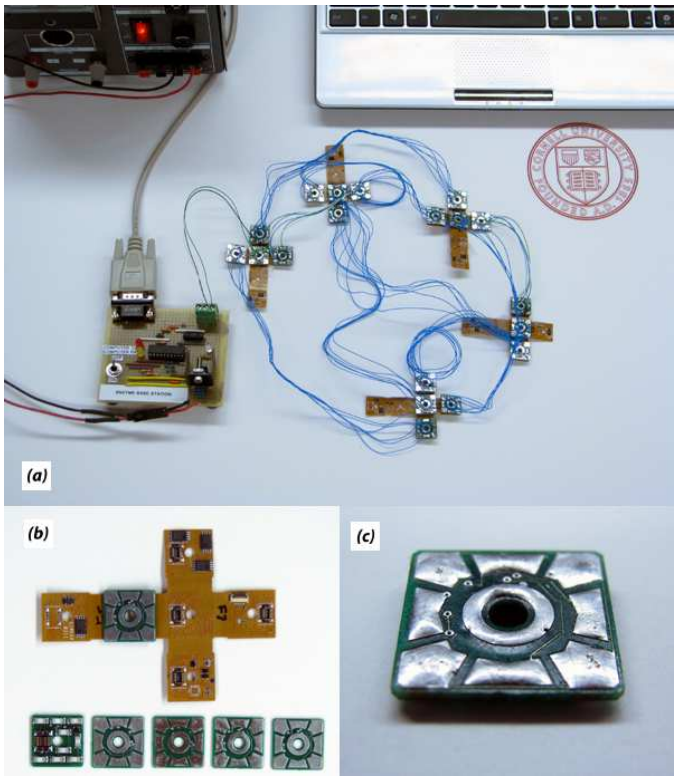


Figure 5.2: Functional validation test of single wire bus communication system. (a) Photo of experimental setup for confirming communication reliability in mesh arrangement. Five module circuit boards are connected through axillary wires to resemble a structure realistically encountered in self-assembly. A PC is connected through an RS-232 converter and a power supply provides power in place of the assembly chamber substrate. (b) The cross shaped circuit boards are in their unassembled state. For assembly they will be folded into a 3D cube and inserted into a module skeleton. (c) Cube face used for connection, with Field's Alloy covered connector pads for use in a self-soldering connector.

chitecture is that voltage levels on the communications bus remain constant after each additional module is added to the structure, and small voltage fluctuations at the time of connection have no effect on the existing structure. For each of the three arrangements shown in Figure 5.3, 20 tests where performed sequentially. Each experiment consists of a sequence of *PG*, *CA*, *SL*, *SL*, *CA* commands being sent to every connected cube. Correct switching of the status LED and correct *CS* messages as reply to each command are evaluated to judge signal transmission. For each arrangement, 100% of sent messages

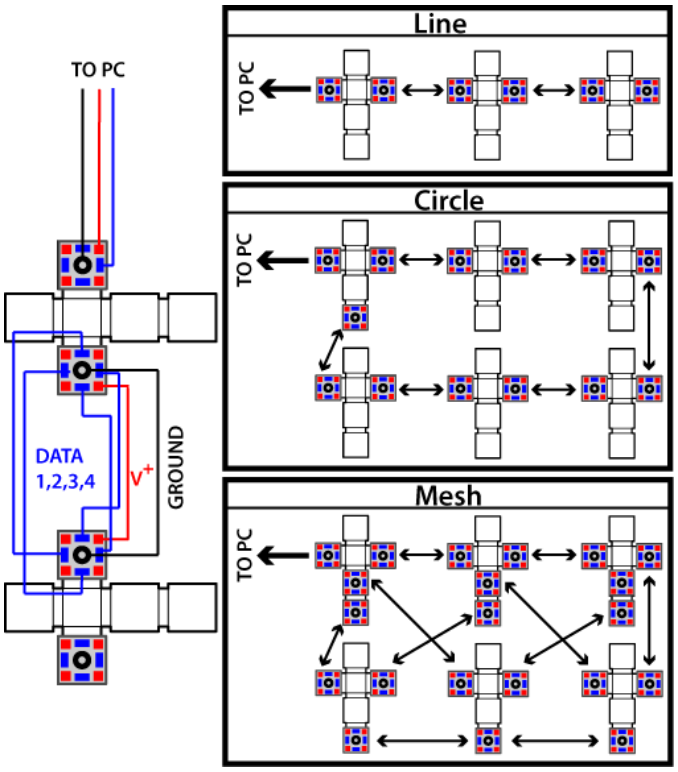


Figure 5.3: Schematics of experimentally validated topologies of module communication networks. Master-slave communication was experimentally validated with cubes connected in a line, loop and mesh. In each case one cube was connected through the RS-232 level converter to the PC from which commands where sent into the structure. Instead of using fully assembled cubes, we connected cross-shaped circuit boards (which will later be folded and assembled to become part of the assembled module) with wires.

resulted in a correct reply received by the PC and the status LED on every cube was switched correctly at every instance. In fact, transmission errors could only be provoked by transmitting a message from the base station concurrently with a message being sent by a module – an expected failure mode.

5.6 Discussion

In this chapter, a review of existing communication architectures used in self-assembling robotic systems led to the conclusion that a single line bus topology using electrical connectors is the most promising option for modular robots with minimally complex modules. The scalability limitations of a bus topology are outweighed by the practically disadvantageous complexity requirements of other technologies.

Subsequently, a detailed description of the implementation of a communication architecture fur such a minimally complex system has been described that is suitable for stochastic fluidic self-assembly. Because the behavior of a single wire bus operating on the UART serial communication pins of a microcontroller may lead to non-intuitive results, the correct function of the proposed implementation has been confirmed experimentally.

The following chapter of this thesis will discuss the stochastic fluidic assembly system alluded to as a potential application for the minimally complex modular robot components including this communication architecture, the thermorheological valve presented in Chapter 4 and the self-soldering connector presented in Chapter 3.

CHAPTER 6

APPLICATION I: A ROBOTIC MODULE FOR STOCHASTIC FLUIDIC ASSEMBLY OF 3D SELF-RECONFIGURING STRUCTURES[†]

6.1 Stochastic Self-Assembly

Of the many existing self-reconfiguring modular robot systems, reviewed in Chapter 2 of this thesis, only very few are capable of self-assembly of robots from disconnected parts. Both the formation of individual modular connections and the acquisition of free modules from the environment remain challenges.

For the *PolyBots* system, for example, an entire paper is devoted to methods for picking up one unconnected module using a chain of modules assembled to form an articulated arm [75]. The strategy applied there and in most other self-assembly demonstration relies on infrared (IR) emitters embedded in the free module's surface. The IR intensity is measured by receivers embedded in the connection counterpart. To pick up a module from the environment, the PolyBot system actuates the modules in the arm to follow the gradient of the measured IR intensity and the chain of robots forming an arm is able to approach the free module and ultimately connect. A similar IR beacon based method is used in for the autonomous docking of two independent multi-module chain robots of the *CONRO* system [135]. The same approach is implemented with cameras and scaled to three disparate robots connecting into one by Yim et al. in a demonstration of re-assembly of a destroyed *CKBot* robot after an explosion [50].

Few more complex self-assembly implementations are reported in literature: Multi-

[†] This chapter includes content previously published in Neubert, J., Cantwell, A., Constantin, S., Kalontarov, M., Erickson, D., Lipson, H. (2010) "A Robotic Module for Stochastic Fluidic Assembly of 3D Self-Reconfiguring Structures", Proc. Int. Conf. on Robotics and Automation (ICRA'10), Anchorage AK, May 2010, pp 2479-2484.

ple external modules are picked up in sequence by a *Molecubes* robot in a demonstration of self-replication [21]. However, these modules are presented at specific locations with precise positioning and alignment. The *SwarmBots* system was used for the most complex self-assembly demonstration with robots to date: 15 unconnected SwarmBots were placed in a 50 cm circle and autonomously connected, also using beacon detection and motion control to approach and connect to other modules [49]. The complex sensor and control systems involved in these self-assembly demonstrations make the self-assembly process unreliable and difficult to scale. In the *Miche* system the problem of performing a guided approach between modules is circumvented by starting from a block of connected modules and performing a disassembly process to reach a target structure [24].

A further challenge in modular robotics is the scalability of systems. The most complex self-reconfiguring robotic system by the number of active modules to date is *ATRON* which consists of 100 modules. This highlights a more general trend in the area of modular robotics: While simulations of large scale modular robotic systems are frequently used to analyze systems of hundreds of modules, implementations of the same systems are hampered by limitations and uncertainties encountered in the physical world [5].

In recent years stochastic assembly systems have received increased attention in the field of modular robotics because they show the potential to overcome both the aforementioned problems. In stochastic self-assembly, no path planning is performed before connecting modules. In fact, all movements of modules including the approach phase before forming a connection are uncontrolled and undeterministic. Instead, modules are brought into contact through a random motion. For each randomly caused contact it is then decided whether the modules bond together or not.

Both the *Programmable Parts* [25] and *Stochastic 2D* [22] demonstrate stochastic

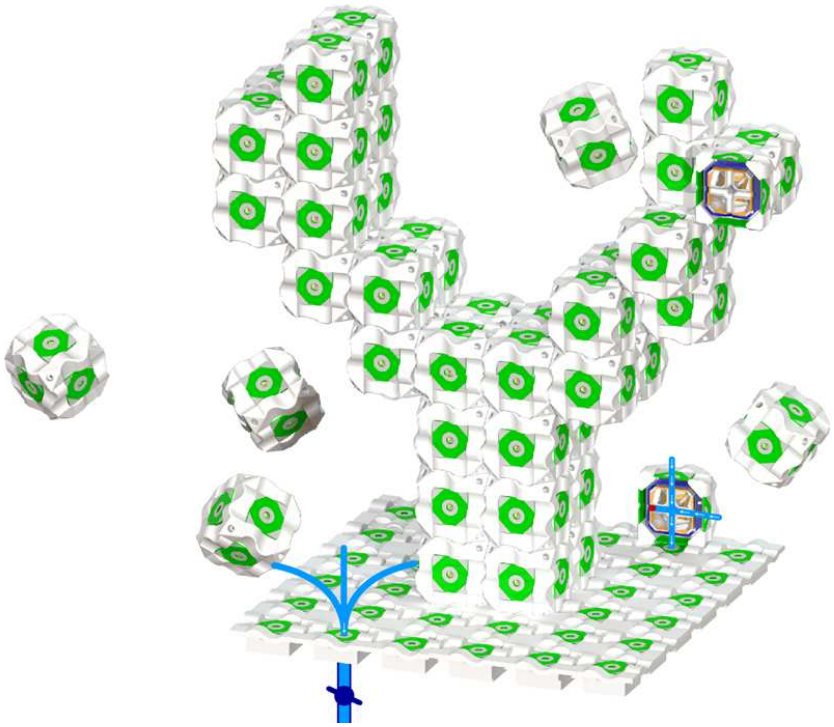


Figure 6.1: Conceptual visualization of 3D stochastic fluidic assembly. Modules are floating in a mixed chamber before being assembled into a target structure. A substrate can create flow sinks to attract cubes to substrate sites (bottom left). Once assembled, the module is powered, and effectively becomes part of the substrate by selectively opening and closing its integrated valves to route flow sinks and sources created by the substrate to its own surface (bottom right). During the assembly process, the order of assembly is not deterministic but guided by the self-assembling structure itself.

assembly in two dimensions: In both systems modules with magnetic connectors are placed on a shaker table. When modules collide such that two sides face each other that are magnetically inverse, a connection is formed, otherwise the modules continue moving randomly. The assembly process is controlled simply by the connector surfaces' patterns. How this approach can be scaled to define rules, or grammers, in order to assemble arbitrary target structures is investigates by Klavins et al [174]. *XBot* is a more recent system that explores the same approach but with deterministic external actuation, resulting in a smaller set of reachable configurations [175].

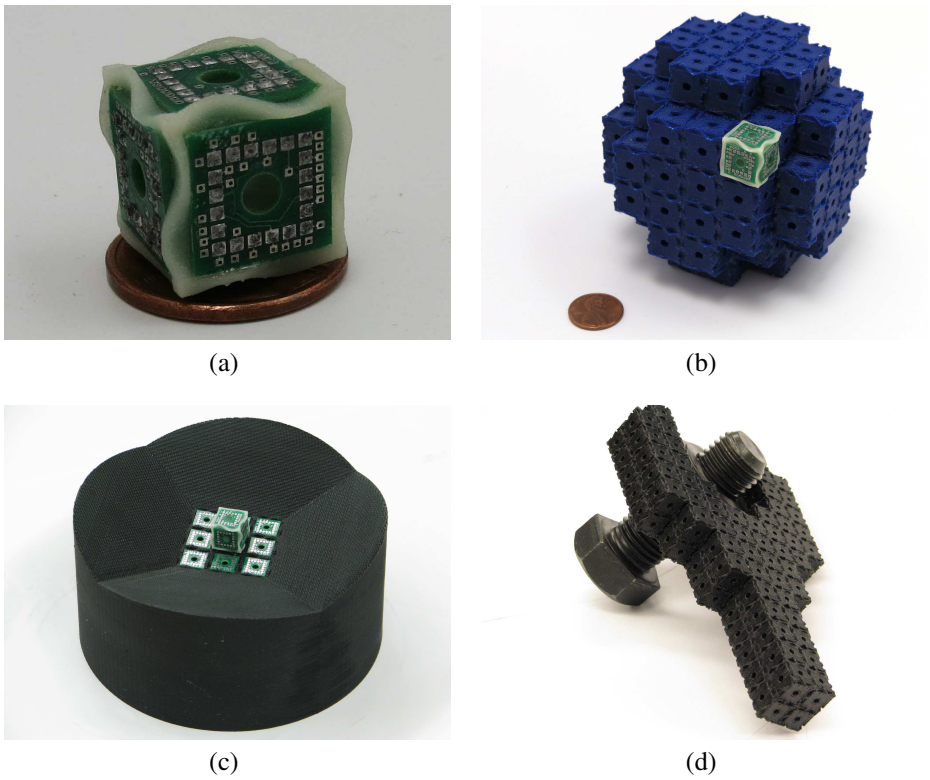
The *Stochastic 3D* system aims to apply the concept of stochastic assembly in three dimensions by placing the cube shaped modules in a mixed fluidic chamber [23] and achieved the manipulation of one module around a fixed seed. Tolley et al implement a similar three-dimensional system with centimeter-scale passive modules and are able to assemble target structures of up to four modules [176].

In order to achieve *self*-assembly modules must have the capability dynamically decide for each random collision whether to form a connection or not, instead of passively. This is possible even when modules are not individually powered: Under the assumption that assembly originates from an initial seed or substrate from which attaching modules can draw power, modules require no independent power supply and can be completely passive. Figure 6.1 provides a concept visualization of a three dimensional stochastic fluidic assembly system.

Krishnan et al introduces the concept of fluidic assembly with tunable affinities where each of the millimeter scale two-dimensional modules is able open or close flow sinks on its surface, effectively allowing it to dynamically attract or reject nearby modules [157]. Attempts to replicate this concept in three dimensions have so far been unsuccessful [94, 125]. In this following section another attempt at implementing a three-dimensional stochastic fluidic self assembly system with active modules is presented.

6.2 Concept

Each assembly step in the stochastic fluidic assembly system presented here occurs locally between a free floating module and an already assembled structure. Initially, a seed is required, which we refer to as the substrate of the assembly chamber. This substrate



(a) (b)

(c) (d)

Figure 6.2: Mockups of output shapes of a stochastic fluidic assembly process. (a) A prototype module of the 10 mm design with 3D-printed curved module shell for self-alignment, internal channels with resistive heaters for thermorheological valve, and six self-soldering connector PCBs (US penny for size reference). (b) A non-functional prototype module of the 10 mm design with a 3D-printed equivalent shape of an assembled spherical structure of modules (US penny for size reference). (c) Early prototype substrate design with nine substrate connector locations. (d) 3D-printed sample shape for illustrating shapes that could be constructed with self-assembling modules.

is powered externally and has computational capabilities in excess of those of single modules even though this is not strictly required. The substrate is shaped in a way that allows one or more modules to connect to it. Figure 6.2 illustrates the assembly system and the types of artifacts it might be able to assembly by means of several mockups of potential target shapes that could be constructed from tens of modules.

All modules are initially free floating in a chamber filled with a fluid, the assembly

medium. Modules have no onboard power supply and are therefore passive while in disassembled state. A propeller causes turbulent motion in the assembly chamber resulting in chaotic motion of all free modules. To start the assembly process, fluid flow is directed through the assembly chamber exiting through ports in the substrate. Agitated by the turbulent motion but directed by the average downward flow, modules will, at random intervals, come into the vicinity of the substrate. Given sufficient proximity of a module and the substrate, the module will self-align and dock to the substrate.

For a flat substrate the number of ports in the substrate is equal to the number of modules which can connect, but other substrate shapes are possible where multiple substrate outlets are combined into "pockets" that accept one module only. Each substrate port or pocket can be opened and closed independently. Through selective opening of ports, a free floating module can be attracted to a desired location on the substrate. Up to this point in the assembly process this resembles the fluidic assembly system with passive modules previously presented by Tolley et al. [176].

Once a module approached and docked to the substrate, an electrical and mechanical bond is formed supplying the module with power and information. Modules are cube shaped with a curved edge as shown in Figure 6.2(a). Their interior contains channels to route fluid flow from and to any of their six faces. Each of their six ports contains a valve so that after docking to the substrate, the module can partake in the selective attraction of new cubes. At all times will the modules in the already assembled structure be powered and have the ability to communicate with each other or the substrate.

Modules with three or less adjacent faces connected to the structure can be repelled from the structure for reconfiguration or disassembly. This occurs by reversing the direction of flow and breaking the bond between the modules.

6.3 Implementation

Two module implementations for the stochastic fluidic assembly concept were pursued and are described separately in the following sections. Both module types share many features and are based on the same overall design concept illustrated in Figure 6.3: Both module types contain an internal channel structure connecting all six module faces. Wrapped around this channel structure is one layer of flexible printed circuit board (PCB) that electrically connects the six surfaces and houses the main controller. A total of 12 rigid PCBs are connected to the flexible main PCB using twelve-pin board-to-board push connectors. Those six facing the interior of the module control the flow valves for routing flow through the module. Those six PCBs facing outward are the connector PCBs of the soldering connectors used for forming connections with neighboring modules and substrate tiles. The assembly is held together by an outer skeleton which forms the curved edges of the cube shaped module, whose role it is to aid alignment of modules during the approach process and prevent connections from forming between misaligned modules.

6.3.1 Module Connection using Fields Metal

Both module implementations use the soldering connector described in Chapter 3 for forming connections between modules and with the substrate. The outward facing connector pads on the outermost PCBs are covered with Field's Alloy as described in Section 3.5.3. Once a cube is available for docking (and held in place by fluidic forces), the heating resistors on the inward facing side of the outermost PCB are heated and result in melting of Fields Metal on the powered cube. After cooling, the solidified Fields Metal forms both a mechanical and electrical bond.

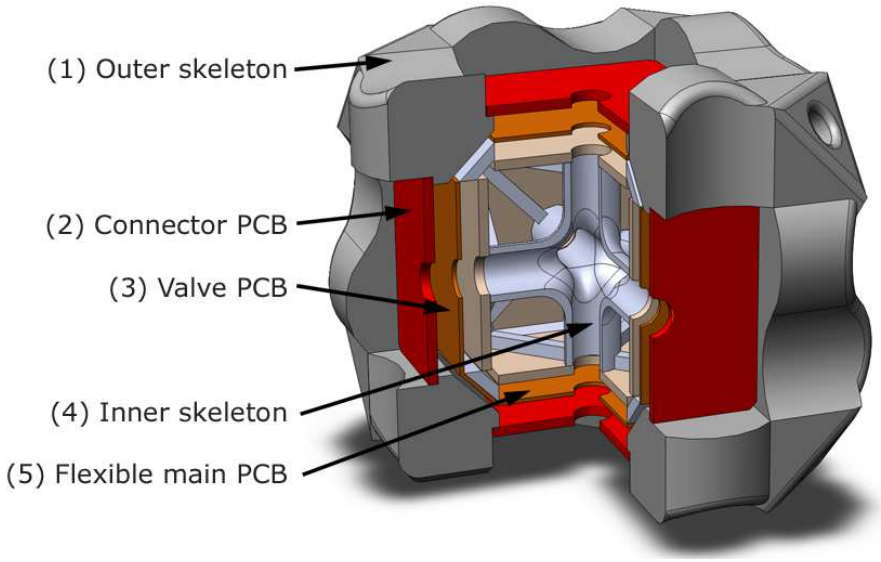
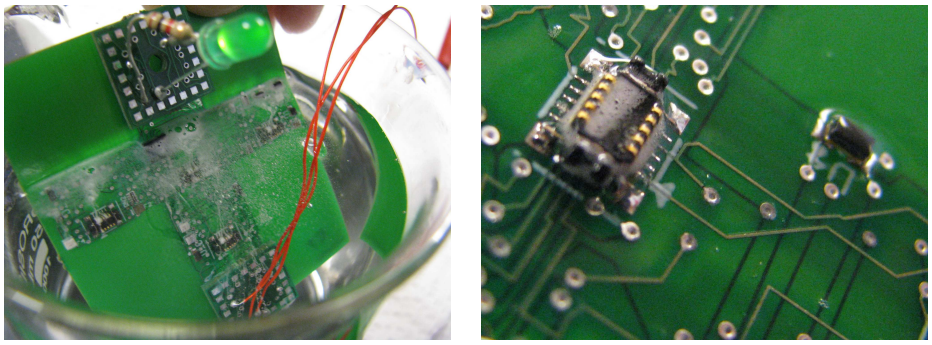


Figure 6.3: Illustration of general design of module for stochastic fluidic assembly system. The cutout makes visible the interior of the module with three layers of PCB as well as interior channels.

6.3.2 Module with Thermorheological Valves

Due to the small size of our modules and space requirements imposed by other functionality, the space available for valves to route fluid flow through the interior of the module is very limited. The port sizes on the module surface are of 2.8 mm diameter, and the height of the valve cannot exceed half the width of our module, i.e. 14.5 mm, but realistically needs to be even smaller to leave sufficient space for mounting and flow routing inside the module. A conventional valve with overall dimensions on this scale is not available, making this an ideal application for the thermorheological valve described in Chapter 4 of this thesis.

Because thermorheological valves work in conjunction with a specific assembly medium that fills the assembly chamber, the compatibility of the thermorheological liquid with all electronics components that were to be used in the module design was investigated. Our trials show that the conductance of all mixtures of Pluronic with deionized

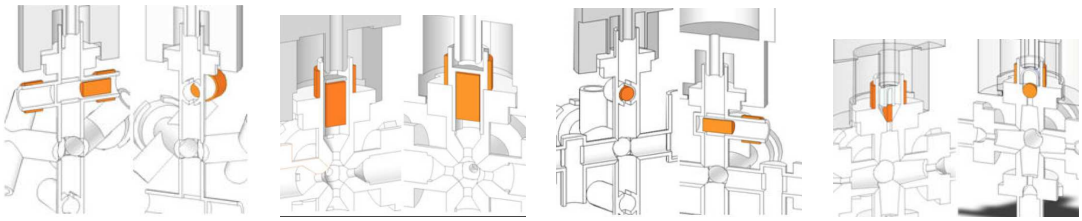


(a) (b)

Figure 6.4: Validation of electrical compatibility between Pluronic® thermorheological liquid and a rigid prototype of the flexible main control PCB. (a) shows the immersed circuit board while in operation with foam forming on exposed current carrying metal surfaces. (b) Closeup photograph of a connector where foam accumulated during submerged operation of the circuit.

water is approximately 100S. Such low conductance provides sufficient insulation of electrical connections to allow immersed circuits to operate normally. Our trials did, however, also show that the timing function of microcontrollers appears to be affected by immersion in Pluronic which was overcome by sealing these with epoxy of type Loctite® Hysol® 120. Further, Figure 6.4 shows the observed foam formation when operating current carrying exposed metal surfaces submerged in mixtures of Pluronic® F127 and Pluronic® F68 thermorheological water. This phenomenon was not investigated further because the circuit remained operational and the foaming was unlikely to interfere with the operation of the assembly system.

The development of the module with thermorheological valves was in parallel with the development of the thermorheological valve itself, as it is described in Chapter 4 and was ultimately not pursued further because the pressures and flow rates that can be controlled with the thermorheological valve are orders of magnitude below those involved in attracting or rejecting modules inside the assembly chamber. As an alternative, a module with six miniature solenoid valves was developed and is described in the



(a) (b) (c) (d)

Figure 6.5: Drawings of selected solenoid valve concept designs. (a) Cylinder magnet travelling orthogonally to flow actuated by two independent coils for opening and closing respectively. (b) Cylindrical magnet in line with fluid flow, requires constant current through coil to overcome fluid drag forces to remain in open position. (c) Cylindrical magnet travelling orthogonally to fluid flow, actuated by one coil for both opening and closing. (d) A disk magnet in line with fluid flow requires constant current through coil to overcome fluid drag forces to remain in open position.

following section.

6.3.3 Module with Solenoid Valves

This alternative module implementation uses the concept of solenoid actuation commonly used for commercially available valves. And while no solenoid valve at the scale required to fit six into a module at the scale of 10 mm is commercially available, there is no fundamental reason why the concept would not work at this scale. The two elements of a solenoid valve is a magnet (moving part) and a coil through which current travels to move the magnet. To reduce the required space for the valve, the magnet itself will directly be inserted into the fluid flow in the valve closed position, as opposed to actuating a separate plunger or using a diaphragm.

Fundamentally, three valve types are possible when attempting to control the flow of a liquid with a magnet: The magnect can be inserted so that the fluid flow pushes

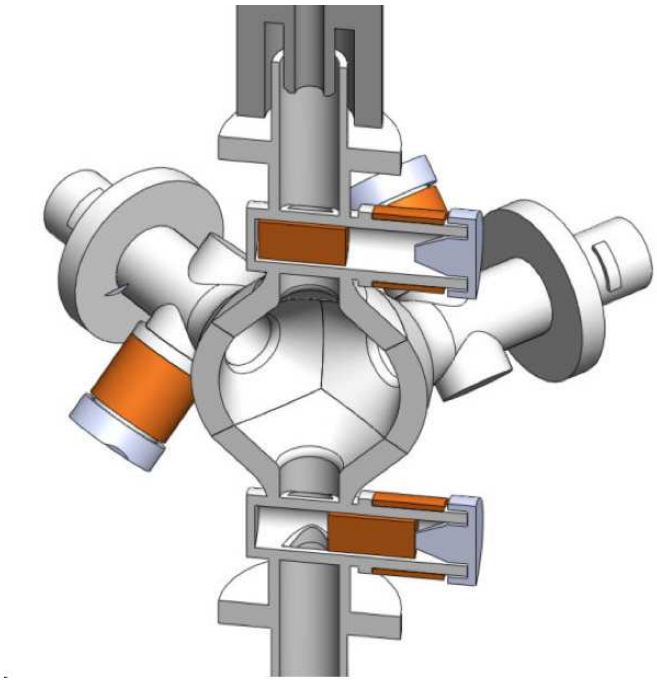


Figure 6.6: 3D rendering of the interior channels of a module for stochastic fluidic assembly. The interior channels of the programmable matter cubes are composed of two identical halves. 3D rendering of cut view show 3D-printed channel structure (white) with magnets and coils inserted (both in orange) and caps to prevent flow from leaking out of the magnet holder.

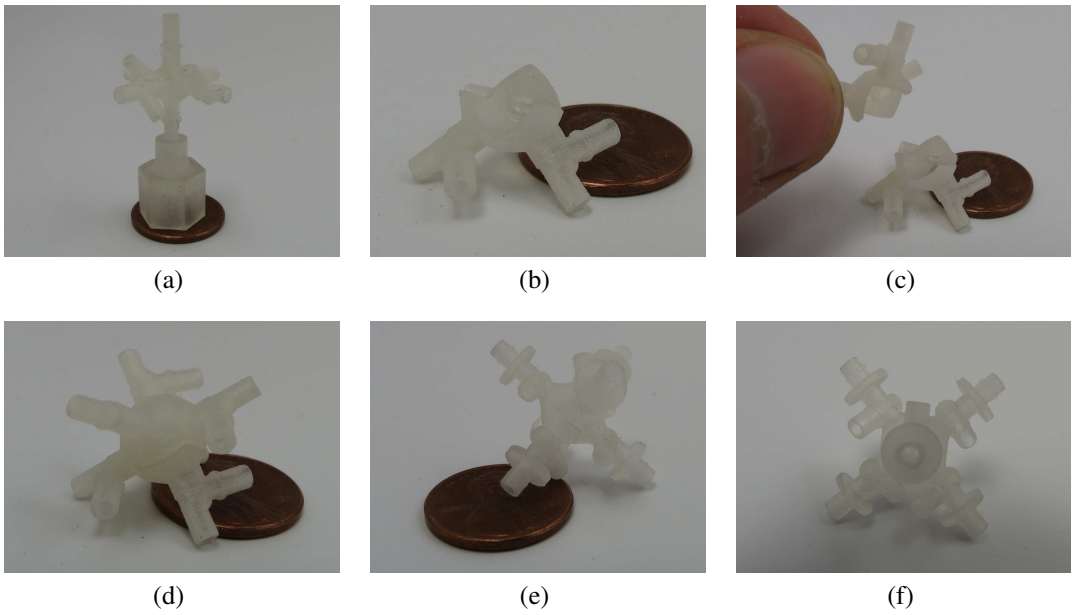
the magnet into the closed position, resulting in a default-closed valve that requires a constant current through the coil to remain in its open position. Conversely, a valve where the flow forces the magnet into the valve's open position requires constant current through the coil to remain in the closed position. The third alternative is a valve where the magnet travels orthogonal to the direction of fluid flow and the drag forces of the flow do neither push the magnet into an open or closed position. This valve design only requires current to be flowing through the coil when the valve is switched, but not to hold it in either open or closed state. Figure 6.5 shows four valve concept, including default closed and orthogonal travel type configurations in the context of the internal channels of a module for the self-assembly system.

After experimenting with several designs that place smaller coin style magnets and

longer cylindrical magnets in line with the flow, all of which failed either due to attraction between different valves' magnets or a too small solenoid force being exerted to overcome the fluidic drag force, we found the only viable solution to be placing the magnet orthogonally to the fluid flow. Furthermore, the orthogonal travel configuration for the solenoid valve is favored over the others because it does not require power to hold either valve position. Figure 6.5(a) shows one configuration with two coils, each driven to pull the magnet for one of the two direction of travel of the magnet. A more space saving configuration is that with just one coil in Figure 6.5(c). The function of the second coil is emulated by reversing the polarity of voltage applied across the coil. This configuration requires careful design of the channel in which the magnet travels: An electromagnetic force only acts on the magnet when it is not or only partially inside the coil. Once the magnet reaches through the length of the coil, it cannot be actuated by that coil anymore.

Figure 6.6 contains a rendered representation of one half of the interior flow channel with magnets and coils. The upper valve is shown in the closed position, while the bottom valve is shown in the open position. Note that the magnet in the open valve position is constrained by a stopper to not move entirely through the coil, to prevent the situation described above wherein it cannot be actuated any more. Figure 6.7 shows photos of the 3D-printed channel which contains wall thicknesses as low as 0.4 mm, the minimal wall thickness that can be reliably printed on the Objet Connex500 printer that was used for fabrication of the channel structure.

Initially, solenoid coils were hand-wound from various thin gauge magnet wire. For later version, the commercially available 75 Ω miniature coil from Plantraco, Inc was used. The coil in combination with a Neodymium magnet satisfies our size requirements and, in experiments, provides enough force to close off or reduce water flow.



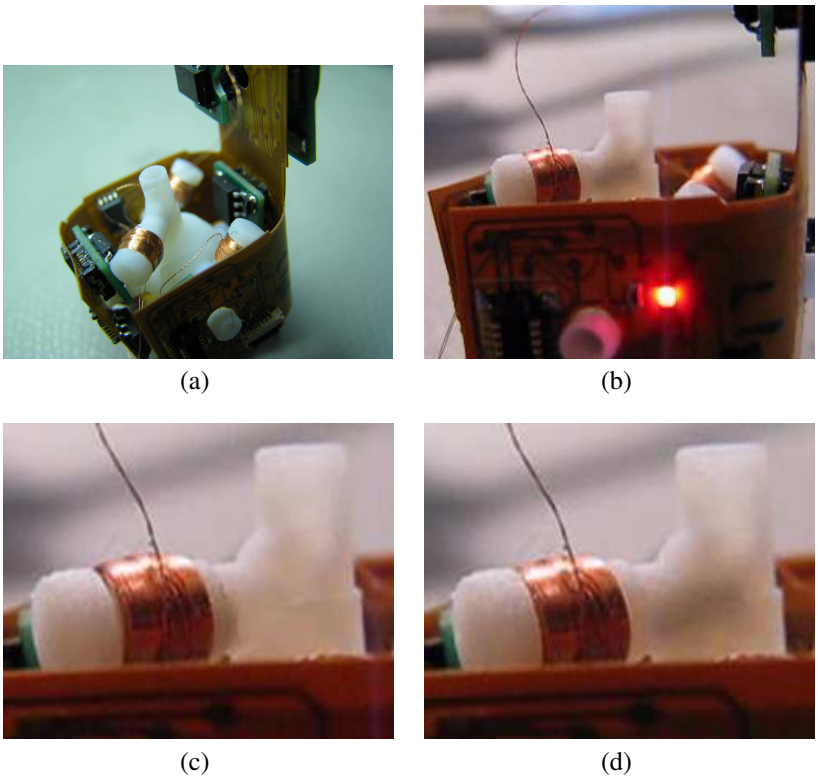
(a) (b) (c)

(d) (e) (f)

Figure 6.7: Photographs of selected internal flow channel designs for the stochastic fluidic assembly module with solenoid valves.

The final chosen solenoid valve design is as follows: A cylindrical Neodymium magnets with a length of $1/8$ in and diameter $1/16$ in travels in a channel of diameter 1.85 mm. Fluid flow is through an orthogonal channel of diameter of 2.15 mm. The magnet is pushed into or retracted from the flow channel by applying a current in the respective direction to the miniature coil of resistance $75\,\Omega$. The channel structure is assembled from two identical halves manufactured by Objet 3D printing using Fullcure 720 material. Plugs, also 3D-printed and designed to snap fit into the magnet housing ends after inserting the magnet, are required to retain the magnet inside the structure. The complete assembly is shown as schematic in Figure 6.6 and embedded into the robot module in Figure 6.8.

Reliability of the switching operation was a concern with early prototypes. Either valves could not be closed reliably or would not open after being closed. This was compounded by support material from the 3D print not being cleaned fully from the interior of the structure and subsequently getting stuck in the small features of the valve.



(a) (b)

(c) (d)

Figure 6.8: Photographs of internal structure of stochastic fluidic assembly module with solenoid valves. (a) Fully assembled cube internals with top surface opened for visibility of flow channels and valves. (b) Side view of same module showing one solenoid coil and the thin AWG40 connection wire between coil and coil controller PCB. (c) Closeup of coil with magnet retracted into fully open position (main flow channel is translucent). (d) Closeup of coil with magnet pushed into fully closed position (flow channel appears dark).

Fabricating the internal cube skeleton as two halves for later assembly allowed us to access the interior after the print process so that all support material can be cleaned out mechanically.

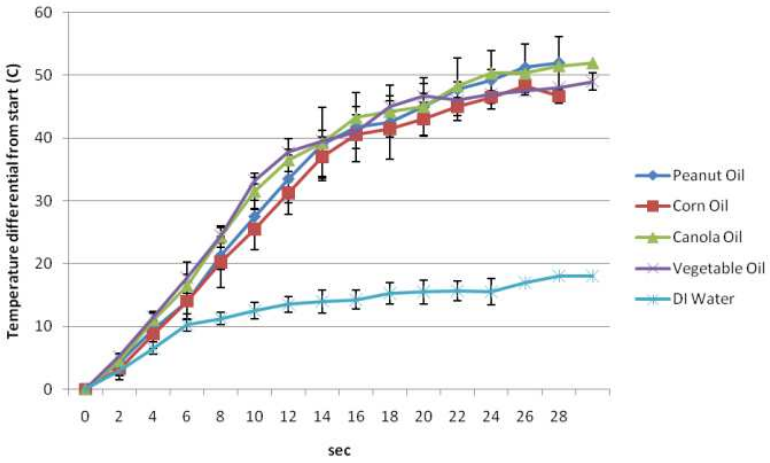


Figure 6.9: Oil comparison experiment: Time history of temperature of one connector pad during the heating process. Sampled every two seconds for four different types of cooking oil and (for comparison) DI water. Temperature is relative to starting temperature of 25 °C. All values averaged over four measurements (except for some values at higher times where measurements were truncated earlier). Error bars represent variance.

6.3.4 Choice of Assembly Medium

In the implementation of the valve that used the thermorheological valve, the assembly medium is prescribed by the findings in Chapter 4. For the system with solenoid valves, thermorheological properties are not required and the assembly medium can be selected based on other properties. One practical considerations are that the assembly medium's viscosity must be low enough for it to flow through the 2.8 mm diameter flow channels under low applied pressures. Another consideration is that the medium should be transparent to allow for visual observation of any assembly experiment. Finally, the power requirements of the soldering connector used to form connections between modules are affected favoribly be less heat conductive assembly media: To effectively heat the Field's Alloy to its melting temperature the power input by the heating element must exceed the energy lost through heat transfer into the assembly medium.

The thermal conductivity for several common everyday and engineering liquids in $W\,m^{-1}\,K^{-1}$ are: Water 0.58, milk 0.57, vegetable oils 0.17, gasoline, kerosine, and mineral oil 0.15. To avoid the problematic handling and disposal of flammable and hazardous substances, the latter three are ruled out. This leaves vegetable oils as a viable assembly medium. Figure 6.9 shows the surface temperature of a Field's Alloy covered connector pad on a soldering connector while submerged in various types of vegetable oil. The heating power is intentionally kept low enough to avoid melting to ensure repeatable tests with the same amount and distribution of Field's Alloy. No significant difference in thermal conductivity is found between different vegetable oils, which led to the selection of the cheapest vegetable oil available to be used to fill the assembly chamber.

6.3.5 Electronic Components

Each module for the self-assembly system contains 13 circuit boards. A flexible PCB wrapped around the internal channel structure of the module provides the backbone of the electrical module design, and is shown in Figure 6.10 in photos and drawings. The flexible PCB holds the main controller, an AtmegaA48 microcontroller, which communicates with other modules and the base station through a communication bus as described in Chapter 5 and controls soldering connectors and valves of the module through its digital output signals. Seven MOSFET devices take these digital outputs as gate inputs to switch the high current loads on the devices. Six N-FET devices are used to select which pair of one soldering connector's heater and one valve is to be switched, while one P-FET device serves to select whether the heater or the valve is activated. This means that either valves or heaters can be actuated at any given time, but the number of either that are actuated is not restricted by the circuit design.

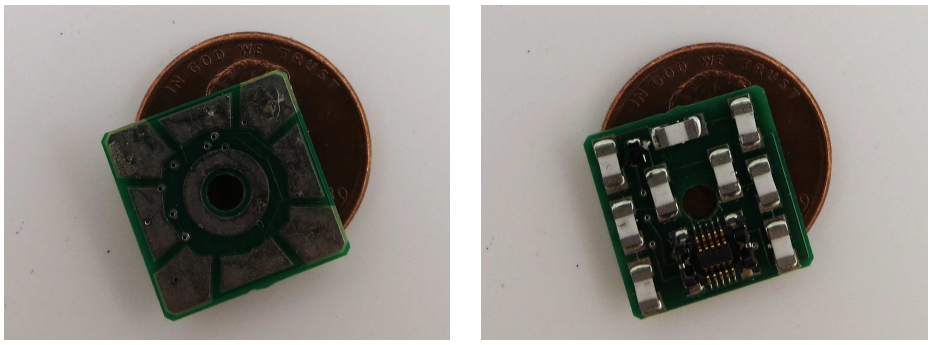
(a) (b)

(c) (d)

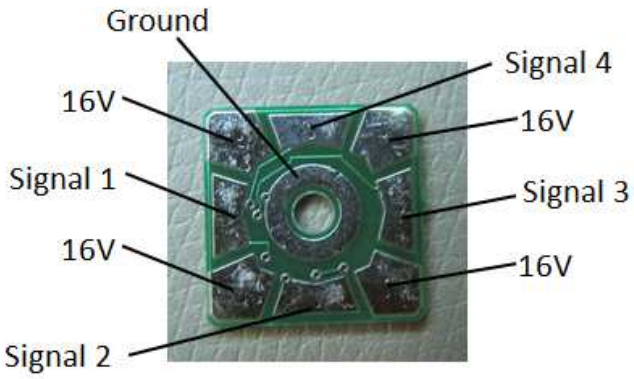
Figure 6.10: Photograph and schematic of main controller PCB. (a) Top view. (b) Bottom view. (c) Layout drawing, top. (d) Layout drawing, bottom.

Other components housed on the main controller PCB are 12 connectors for connecting the breakout boards described below, one programming connector for updating the program on the microcontroller, and several peripheral resistors and capacitors as required by the specifications of the microcontroller. The PCB is designed for adding an accelerometer which was not included in most prototypes to reduce complexity of the assembly process.

The outward facing connector PCB is designed following the guidelines for design-



(a) (b)

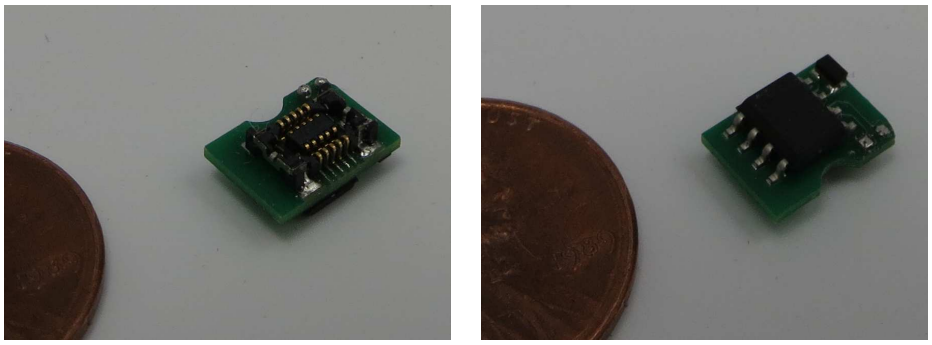


(c)

Figure 6.11: Photographs of soldering connector PCB. (a) Top view of connector pads before applying Field's Alloy. (b) Bottom view of the resistive heater arrangement for melting the Field's Alloy during connection and disconnection. (c) Rotation invariant connector pad electrode layout.

ing soldering connectors in Chapter 3. It contains nine 30 Ω heating resistors, arranged as three parallel sets of three in series to achieve an effective resistance of 30 Ω across the complete resistor array. The PCB is industry standard fiberglass circuit board with a thickness of 1.6 mm. The board dimensions are 13 mm x 13 mm with a hole of diameter 2.8 mm in the center as shown in Figure 6.11. The entire module is supplied with 12 V resulting in a power consumption of 4.8 W when one soldering connector is heating.

The breakout PCB to control the operation of each solenoid valve is shown in Figure 6.12. The PCB is very simple, containing only an H-bridge and a diode to convert



(a) (b)

Figure 6.12: Photographs of solenoid controller PCB. (a) Top view. The upper side of the PCB is occupied by the male side of a 12 pin board to board connector for connection with the flexible main controller board. (b) Bottom view. The side of the PCB facing the module's interior houses an h-bridge and a diode.

the digital signal from the microcontroller into a current of appropriate polarity through the solenoid valve. The fragile wires leading to the solenoid valve, whose thickness is specified as American Wire Gauge (AWG) 40, are soldered into two plated through holes on the solenoid control board after the enamel coating on them has been removed by heating.

Connector selection for forming the connections between the breakout PCBs and the flexible main PCB is challenging due to the limited space availability yet high current rating required. The final choice of part number AA01A-P010VA1 by JAE electronics with 12 connectors was driven purely by the space constraints.

All PCBs were assembled in house using solder stencils and a toaster oven for reflow soldering.

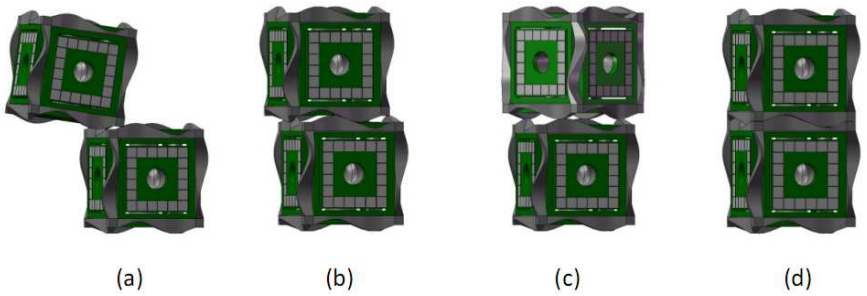
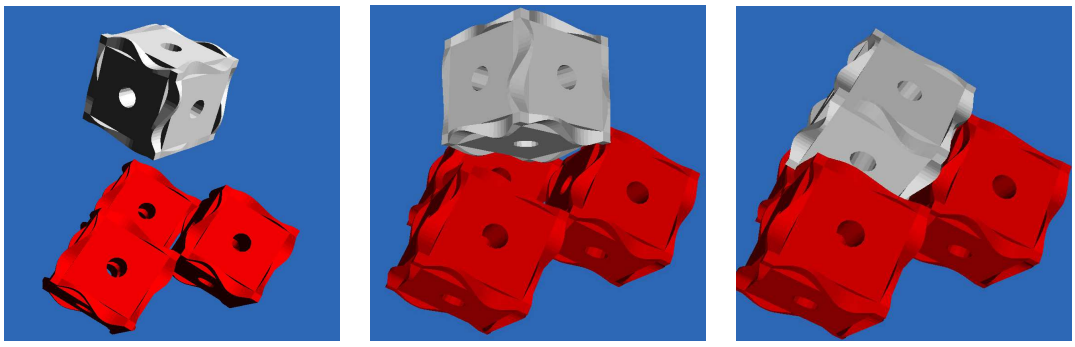


Figure 6.13: Illustration of possible misalignment scenarios during approach phase between two modules. (a) Large translational offset that cannot be correct by curved cube shell. (b) Small translational offset that can be corrected by curved cube shell. (c) Large rotational offset that cannot be correct by curved cube shell. (d) Small rotational offset that can be corrected by curved cube shell.

6.3.6 Self-Alignment During Approach

When relying on stochastic processes one cannot expect modules to arrive in the exact orientation and location as required. Misalignment is expected when one module approaches the already assembled part of the structure and during the docking process.

The solution to the self-alignment problem is a mechanical design in which self-alignment is inherently prevented by means of its shape based on our previous experience with modular systems on a larger scale. Self-aligning cubes feature curved edges which, due to their rotational symmetry about the center of every face, only mesh when the two cubes are positioned and oriented in the correct fashion (see Figure 6.13). While it would be favorable to extend the wavy pattern over the entire surface of a cube face, the need for the contact pads to be attached to a circuit board requires this surface to be flat. The curved edge further prevents contact between the two face circuit boards before the two (or more) connecting modules are fully aligned. For linear misalignments of up to half a cube's side length and all radial misalignments, the curved edge guides the modules into alignment under forcing by the fluid flow.



(a) (b) (c)

Figure 6.14: Screenshots of the approach of one module to a "pocket" formed by three other modules simulated using the Vortex physics simulation engine.

Secondly, the shape of the already assembled structure plays a role in how well the self-alignment process can be guided. Similar to the list probabilities of approach between two stochastically moving groups of modules by Klavins et al [177], one could generate a list of the probability of correct self-alignment between a free cube and different structures. For example, it becomes quickly apparent in trials, that self-alignment occurs much more readily when a cube approaches a "pocket" of three surfaces, than when approaching only a flat surface. While one cannot guarantee such situations throughout the assembly process, we chose to give the base of our chamber from which assembly starts, such a preferred shape as shown in Figure 6.17.

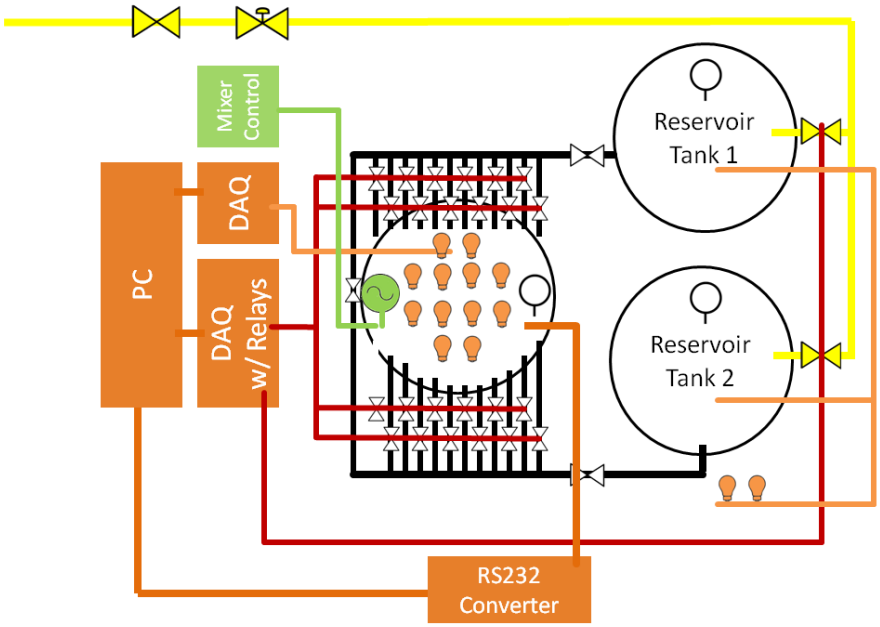
In addition to the considerations described above, the computational physics simulation engine Vortex was used to simulate the approach of one module towards a "pocket" formed by three modules, representing the substrate shape pattern. Screenshots illustrating the simulation setup are shown in Figure 6.14. These investigations remained inconclusive because the results proved to vary greatly with the simulation's friction model parameters whose realistic values are difficult to determine.

6.3.7 Buoyancy

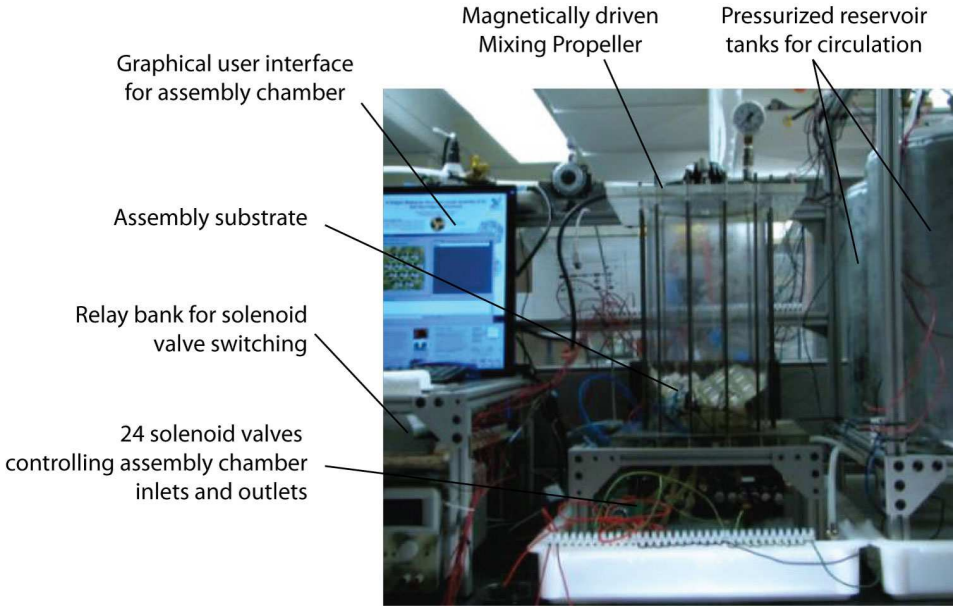
Control of the module's buoyancy is critical to successfully manipulating the module inside the fluidic assembly chamber. To reduce the average density of the module the voids in the module's interior are filled with Glass Bubbles^{TM} K1 by 3M, Inc. This product consists of air filled glass spheres with an average diameter of 1 μm diameter which are commonly used to control buoyancy in underwater applications. To prevent leakage of the glass bubbles out of the module all joint surfaces on the module's exterior, particularly those between the module shell and PCBs, were sealed with nail polish.

6.3.8 Assembly Chamber

Figure 6.15 shows the assembly chamber, both as a schematic and annotated photograph. The assembly chamber is an acrylic tube terminated with two acrylic end plates that are mounted with two 3D-printed seal rings. The lower acrylic plate contains 12 flow ports as well as a sealed opening for all electrical connections to the substrate of the assembly chamber. Mixing of the chamber is provided by an aquarium pump for large aquariums. Each of the 12 assembly substrate sites is connected through one of the twelve flow outlets to two 120 V AC powered solenoid valves controlling the connection to the high proessure or low pressure end of the system. This is how each substrate site can be turned into a flow inlet or outlet. Circulation is not provided by a pump, because trials with various pump types resulted in strong bubble formation and reduced control over flow rate and pressure in the system. Instead, the assembly medium is continuously pumped between two reservoir tanks. At any given point one reservoir is pressurized by connecting it to a pressurized air supply line, while the other is open to atmosphere. Once the latter is filled with assembly medium, the air pressure is switched and flow



(a)



(b)

Figure 6.15: Schematic and photograph of the assembly chamber and supporting components. (a) Schematic. Yellow: pressurized air supply. Black: Vegetable oil circulation. Red: 120 V AC power. Orange: 5 V TTL level signals. (b) Photograph of assembly chamber.

reverses. By simultaneously reversing the function of the solenoid valves between outlet and inlet control, the change in flow direction has no effect on the flow inside the assembly chamber.

6.3.9 Graphical User Interface

Figure 6.16 shows three views of a graphical user interface (GUI) used for controlling the assembly tank substrate and communication with modules through the communication bus.

The GUI and underlying logic are written in C# and implemented as Windows Forms application. Communication with the robot communication bus is through the PC serial port and an intermediate voltage level converter, that converts the PC's serial bus voltages to TTL logic level voltages used on the communication bus, without altering the signal.

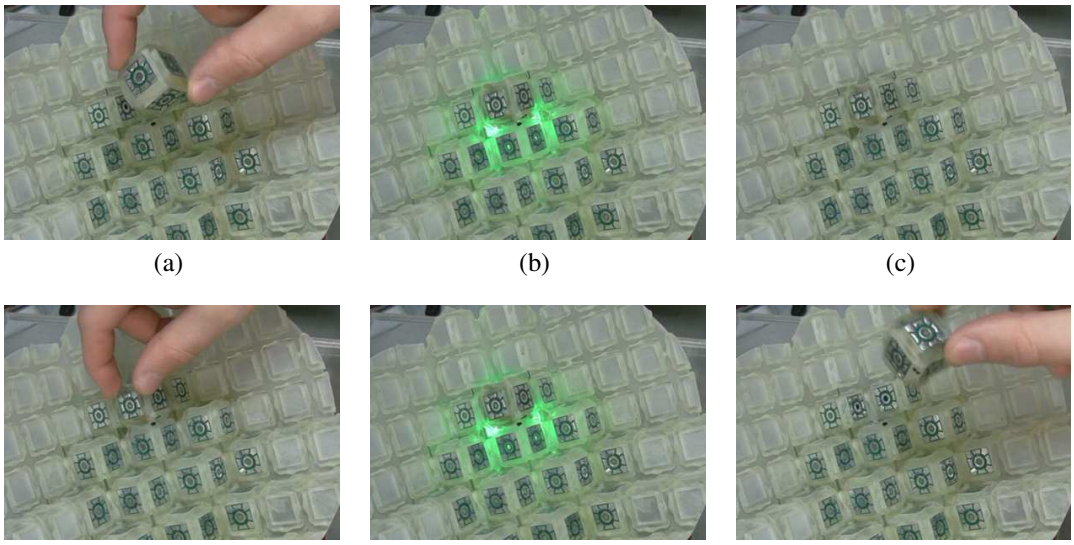
6.4 Functional Validation

6.4.1 Connection

For successful self-assembly it is important that the mechanical and electrical bonding between modules is reliable enough that every cube is connected in a functioning fashion to at least one neighbor. The fundamental functions of the soldering connector connection method have already been validated extensively in Chapter 3. A functional validation of the soldered connection between the substrate and a module (which

Figure 6.16: GUI for control of the stochastic fluidic assembly chamber substrate. (a) Dialog for establishing connections to the communication bus of the modular robot system through the PC serial port and to the digital I/O device for substrate valve control. (b) Message building interface for constructing messages to be sent to robot modules via the communication bus. (c) Valve control interface for controlling all 24 substrate valves as well as visual LED indicators embedded in the substrate.

is equivalent to connections between modules at later stages during the assembly process) was performed by manually placing a module in the fully assembled 12-position substrate. Figure 6.17 shows the experiment in a sequence of photographs: In Figures 6.17(a) to 6.17(c) the module is placed and connected (the green illumination is to indicate the soldering connector being powered). Figure 6.17(d) shows the test of attempting to manually pull and shake the module to break the connection, which fails.

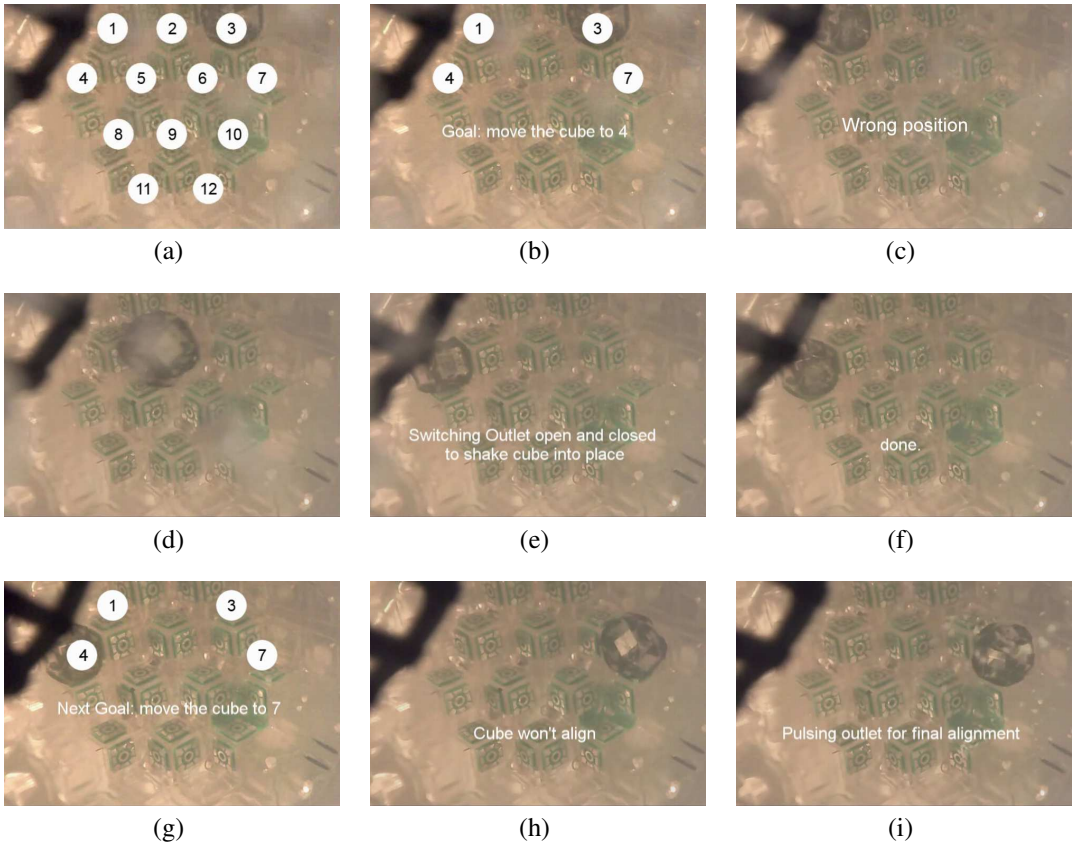


(a) (b) (c)

(d) (e) (f)

Figure 6.17: Test of module conntection outside assembly chamber. (a) A module is manually placed in one of the assembly chmaber substrate's connection sites. (b) The soldering connectors of the substrate site are activated, the illumination is for visual indication of the heater state. (c) After heating the connection is left to solidify for 30 s. (d) The strength of the connection is check by manually pulling on the module. (e) To break the connection the soldering connector is heated again. (f) The module can be manually removed by pulling.

Subsequently the soldering connector in the substrate is simply activated again and the module easily removed in Figures 6.17(e) and 6.17(f). During the test, module successfully seated itself in the substrate "pocket" once the Field's Alloy was molten and received power as indicated by the blinking status diode. As a further confirmation of correct functionality of the module and connection method, messages sent to the module through the communication bus were acknowledged with the appropriate response. This shows that the overall cube layout and substrate design are suitable for forming larger assemblies.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.18: Sequence of photographs of attempted module manipulation in stochastic fluidic assembly chamber. (a) Photographs are taken through the top of the assembly chamber showing a top view of the assembly substrate with 12 individually controlled substrate ports. (b) Only ports 1, 3, 4, 7, are active in this experiment. A passive module starts at port 3 and the human operator attempts to move it to port 4 only by actuating the valve that control flow through the four active substrate ports. (c) After a first attempt the module comes to rest on substrate port 1 which is adjacent to the target position. (d)-(f) The operator switches port 1 to be a flow outlet, ejecting the module and starting manipulation again, this time resulting in the module coming to rest above substrate port 4. (g) After switching port 4 between closed and inlet configurations repeatedly the module comes to rest in port 4. (h)-(i) Subsequently, the operator attempts to manipulate the module to move to substrate port 7.

6.4.2 Module Manipulation in Assembly Chamber

To investiage whether the concept of manipulating modules in a mixed chamber through forming flow sources and sinks results in the expected acquisition of modules in target locations, a passive module with a buouancy approximately equal to the fully function module was inserted into the assembly chamber. The flow valves connected to the assembly chamber substrate's flow ports were controlled manually using the GUI described in Section 6.3.9.

Figure 6.18 shows a sequence of photograph taken throughout the experiment. Figure 6.18(a) shows the location and numbering scheme chosen for the flow ports in the substrate with the module located on port 3 in the top right of the image. Through manual manipulation of the valves the human operator then attempts to move the module to come to rest on top of substrate port 4. Figure 6.18(c) shows the module approaching port 10 on the other side of the substrate instead. After several further attempts (not shown) the module does eventually approach port 4 in Figures 6.18(e) and 6.18(f). Figures 6.18(g) to 6.18(i) show the subsequent attempt to move the module to port 7 which is successful but the module does not move fully into the port and comes to rest above it. The human operator overcomes this problem by switching the flow port between inlet and outlet at a frequency of approximately 2 Hz effectively shaking the module into place.

This experiment shows that while possible, the manipulation of modules in the assembly chamber is difficult for human operators and likely difficult to automate.

6.5 Conclusions

This chapter described two implementations of a system for stochastic fluidic modular assembly where the assembled structure itself controls the assembly process while energy is supplied externally.

Components of the systems were only validated in isolation and while all necessary components, including the module and the assembly chamber, were constructed and their functionality demonstrated, the integration of the complete systems was not pursued to the point where multiple modules could be assembled. The primary challenge encountered in scaling the system to support full assembly of multi-module structures remains in designing modules which can be manipulated with fluid flows of similar magnitudes as those flows the modules can modulate. In our current implementations the pressures and flow rates applied at the assembly chamber substrate in order to manipulate the module were multiples of those that could feasibly be controlled with the two types of miniature valves developed for the interior of the module.

Possible directions for further work on stochastic fluidic assembly systems might be found in continued development of the miniature flow switching techniques such as the ones used in our implementations. Alternatively, or in conjunction, interaction between the design of modules, the selection of the assembly medium, and the assembly chamber design could be further investigated to gain a better understanding of how the stochastic process of mixing the assembly medium can be better controlled towards the desired outcome.

CHAPTER 7

APPLICATION II: AN ECOSYSTEM OF MODULAR SELF-REFINING MACHINES[†]

7.1 Concept

In ecology, the concept of an ecosystem encompasses the network of interactions between a group of organisms and their environment. Broadly speaking, ecology studies the flow of information, energy, and matter through the network. Food networks, or food chains, are common examples for network interactions in ecosystems: Organism acquire inorganic and organic matter from other organisms and the passive environment and use it for biological functions such as metabolism, growth, self-repair, and self-replication.

Biology frequently provides inspiration for robot design and some feature of ecological networks have already been demonstrated in robotic systems: The flow of information among groups of robots is an extensively studied topic commonly described under the term *swarm intelligence* [37, 178]. The collective search and distribution of energy in a robot ecosystem has been demonstrated by Belpaeme et al [179]. Modeling the flow of matter in a robotic system has so far been the most elusive aspect of transferring the concept of ecosystems to machines. Nigl et al. [32] demonstrate machine metabolism with an assembler robot that acquires and rejects truss components from the substrate it exists in. Zykov et al. [20] demonstrated self-replication in a modular self-reconfiguring robot that through interactions with the environment acquires modules and constructs a replica of itself.

[†] This chapter includes content previously published in Neubert, J., Lipson, H. (2014) "Soldercubes: A Self-soldering Self-reconfiguring Modular Robot System", Autonomous Robots, submitted. This chapter also includes content previously published in Neubert, J., Rost, A., Lipson, H. (2014) "Self-Soldering Connectors for Modular Robots", IEEE Transactions on Robotics, submitted.

The exchanges of information, energy and matter are inherent functions of self-reconfiguring modular robots which makes them a suitable hardware for demonstrating ecological interactions in engineered systems. In this chapter a self-reconfiguring modular robot system is presented that is designed for the purpose of implementing complex physical ecosystem interactions between modules and several demonstrations of such interactions are presented.

7.2 The *Soldercubes* Self-Reconfiguring Modular Robotic System

The modules of the Soldercubes modular robot system are designed to be the basic building blocks of modular machines that exchange modules with their environment. The design requirements were not defined on a per-module basis, but by defining functions and scenarios that robots assembled from arbitrary numbers of modules must support. These basic required functions are: (1) Acquisition of modules from the environment into a robot, (2) rejection of modules from the robot into the environment, and (3) manipulation of modules. The scenarios that must be supported are: (1) Growth of robots through the acquisition of modules, and (2) self-improvement of robots through the acquisition and rejection of modules or assemblies that lead to extended functionality.

7.2.1 High-level Design Considerations

The primary design constraint for the module design is that tens of modules must be built to realize even the most basic demonstration of a machine ecosystem. Few self-reconfiguring 3D modular robot systems reported in literature have constructed more than 10 modules: Only Polybot with 56 modules [5], and ATRON with 100 modules [54]

currently exceed this threshold[1]. To improve scalability, the reduction of complexity, material cost, and assembly time for every aspect of the module design were considered, while still enabling the feature set necessary for the desired interactions between modules. The following describes these considerations in detail.

Connection Mechanism A large number of connectors per module greatly increases the flexibility of a modular robot system. Each additional connector also contributes to the cost and complexity of each module. Chapter 3 provides a detailed discussion of methods for forming connections between modules without external manipulation and describes a novel self-soldering connector. This connector meets all requirements for self-reconfiguration, is self-contained on one circuit board and contains no moving parts. Due to its simplicity the soldering connector is used for demonstrating the concept of robot ecosystems in this chapter. Because this connection method is a unique trait of the modular robot system presented here, the name "Soldercubes" is used to refer to the system from here on.

Dimensionality While two-dimensional systems provide the benefit of design simplicity and free the designer from having to consider the effects of gravity, 2D systems do not easily apply to real-world applications. In order to implement meaningful demonstrations of growth and robot interaction, Soldercubes are a three-dimensional system.

Topology Type To avoid the difficulty of arbitrarily aligning modules in 3D space, Soldercube modules are constrained to only interact with other modules when aligned with a 3D grid. This inherently makes Soldercubes a *lattice type* modular robotic sys-

[1] In 2D systems the largest number specified in literature is 24 Crystalline modules [79]. The only mass produced modular robot system is the manually reconfigurable Cubelets system with 55,026 modules produced by October 31, 2013 [121].

tem. This is, however, not a fundamental constraint of the design: If the problem of precise 3D positioning is solved, for example through external sensor inputs, Soldercubes can be configured and used in chain type configurations. Additionally, Section 7.7.3 discusses hardware extensions towards mobile robots assembled from Soldercubes.

Number of Actuated Degrees of Freedom Most recent lattice type self-reconfiguring modular robots with actuated degrees of freedom (dof) contain one or 1.5 rotary actuators per lattice cell, while previous systems attempted to house up to six actuated dof per lattice cell and some explored linear actuation (see Table 2.1 for an overview). The trend towards smaller dof count per lattice cell is likely due to the broad availability of increasingly power dense servo motor actuators and the insight that large numbers of simple modules are more likely to meet the goals of versatility, low cost, and reliability than fewer complex modules. Following the biological inspiration in demonstrating a robot ecology, the Soldercubes design follows the same approach to minimize the complexity of each module.

Placement of Actuated dof Several options for the orientation of the single actuated dof within the module have been explored: In the Molecubes system, the actuator is aligned with the long axis of the cubic lattice cell [21] rotating equal halves of the module relative to each other. The Cubelets actuation module rotates one face of the cube shaped module relative to the other five faces. CoSMO sacrifices 2 out of 6 possible connector faces to allow two L-shaped halves to change their relative positions through rotations. The Roombot and Superbot systems both use metamodules extending over two adjacend lattice cells containing three actuated dof to remove the bipartion inherent to 1-dof systems[2]. Soldercubes follow the Cubelets design which, like Molecubes, al-

[2]In systems with one rotary dof per module, space is divided into two halves: Lattice cells can be thought of as arranged in a three dimensional checkerboard pattern. Modules in "black" lattice cells are

lows for all six cube faces to be used as connectors but avoids the complex motion paths caused by rotation axis of the Molecubes actuator.

Actuator Selection Because Soldercubes only interact when aligned with a 3D grid, only rotations in 90° increments are required. To demonstrate interactions between robots, lifting module assemblies will be required for locomotion, for which the torque produced by the motor needs to be considered. High torque output alone is not sufficient, however, as this normally goes along with larger motor size. Table 7.1 lists several motor packages that were considered. For each option, the torque characteristic and approximate resulting module size are used to compute how many modules could be supported in a cantilever configuration at stall. Besides the actuation requirements, a position sensor is required for positioning. Several options in Table 7.1 are servo motors of the Dynamixel series by Robotis Inc, which include a potentiometer and integrated motor controller. When selecting a servo motor package with potentiometer, one must consider whether the actuator supports continuous rotation, and the size of the dead band of the potentiometer. Continuous rotation reduces the constraints on robot motion significantly and later sections of this chapter will in fact illustrate that it is required for legged locomotion of Soldercubes robots. The dead band of the potentiometer is the region in which the potentiometer does not produce a sensor reading. With the Soldercube module requiring positioning to four equally spaced positions, the dead band must be smaller than 90°. The final actuator selection is for Dynamixel AX-12A by Robotis that meets the basic requirements, includes a potentiometer and provides an acceptable tradeoff between cost and torque output.

not able to move to “white” cells and vice versa. [88], [170]

Table 7.1: Servo Motor Selection Options

Product Name	**price ($)**	**incl. encoder**	**gap**	**weight (g)**	**torque (Nm)**	**module size** (mm)	**cantilever (length)**
Sparkfun "Large"	12.56	✗	N/A	44	4.32	50	7
Sparkfun "Medium"	10.78	✗	N/A	18	1.77	45	10
Dynamixel AX-12A	42.66	✓	60	55	5.39	60	9
Dynamixel AX-18A	90.16	✓	60	54.5	5.34	60	9
Dynamixel DX-117	179.10	✓	60	66	6.47	50	14
Dynamixel RX-28	199.41	✓	60	72	7.06	60	12
Dynamixel MX-28	219.90	✓	0	72	7.06	60	9
Dynamixel RX-24F	132.91	✓	60	67	6.57	60	10
Dynamixel RX-64	265.91	✓	60	125	12.26	70	11
Dynamixel EX-106	474.05	✓	109	155	15.20	75	12

Heterogeneity Soldercubes are designed to be operational as a system of only actuation modules. To save cost and assembly labor, complexity of the system can be further reduced by introducing heterogeneity with structural modules that have the same shape and interface as actuation modules but no actuator. Soldercubes are extensible with further module types as long as shape-homogeneity is preserved and Section 7.7 explores possible hardware extensions.

7.2.2 Actuation Module

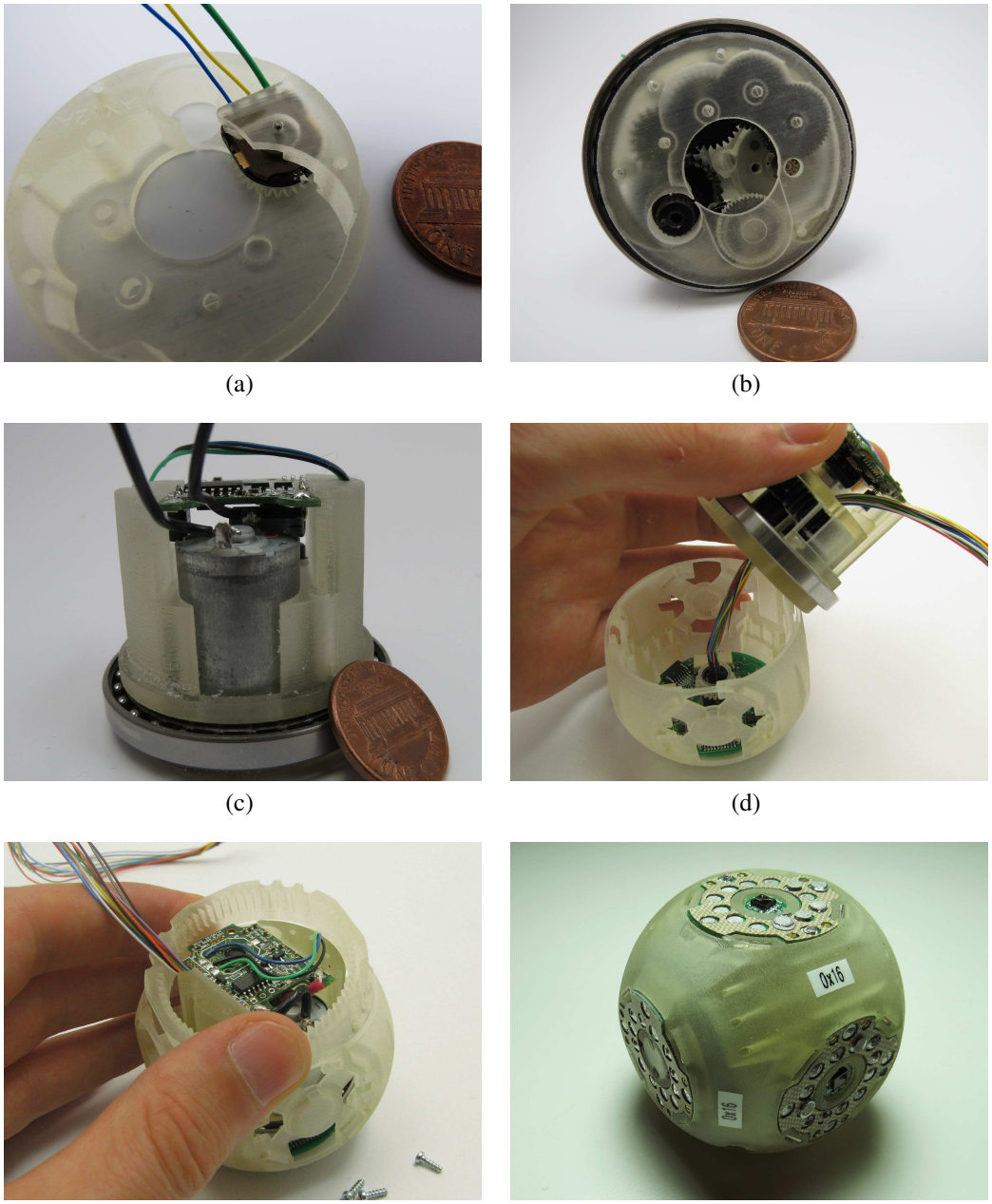
Considering the space requirement of the actuator and other components that are needed to enable relative motion of one part of the module with respect to the other, the actuation module has most design constraints, and was designed first.

The actuation module is split into two sub-assemblies which are actuated to rotate relative to each other: One side contains a shell with five of the cube's outward facing connectors. Mounted inside this shell are the module's main controller PCB, as well as a large internal gear around the circumference of the interior space of the shell. The

other side has only one external connector and is connected to the motor and gearbox assembly which fill most of the space inside the module. The two halves' only points of contact are a ball bearing and the contact points of the large internal gear with a smaller gear protruding from gearbox assembly. When actuating the servo motor, the gearbox assembly rotates inside the cube shell body.

It is essential to the operation of any robot assembled from Soldercube modules that the single cube face which rotates relative to the others contains a connector for other modules to connect. Therefore, power, the communication signal, and the control signal for the connector must be connected between the two parts of the actuation module even though they are free to rotate infinitely relative to each other. To enable this feature the Soldercubes actuation module contains a Moog SRA-73540-12 slip ring with twelve contacts. Each of the twelve contacts rated at 2 A, eight of which are reserved for power transmission, allowing for up to 8 A to be transferred through the rotating joint inside the actuation module. Based on experimentally measured power consumption of module functions, this means that up to eight connectors of neighboring modules could be operated at the same time while being connected through the joint in this module only. Of the remaining four wires in the slip ring, one is used for the communication signal bus, one for controlling the connector on the rotating module part away from the main control PCB, one for sending commands to the servo controller PCB, and one for the adjacency sensor signal. The space requirements of the already small slip ring package are further reduced by removing the outer shell of the plastic housing and using a razor blade to remove excess wire protection. A small cap with an external spur gear profile is press fit onto the interior housing of the slip ring to prevent the slip ring assembly from falling apart.

While the servo motor package of the Dynamixel AX-12A product is convenient for



(a) (b) (c) (d) (e) (f)

Figure 7.1: Photographs of actuation module assembly process. (a) Potentiometer enclosure cap: The top part of the servo motor assembly houses the potentiometer (b) Servo motor assembly, top view. (c) Servo motor assembly, side view. (d) The wires connecting the rotating parts are routed through the center of the gearbox. (e) Inserting a 3D-printed internal gear which acts as an additional reduction stage and clamps the ball bearing into the cube shell. (f) Completely assembled actuation module.

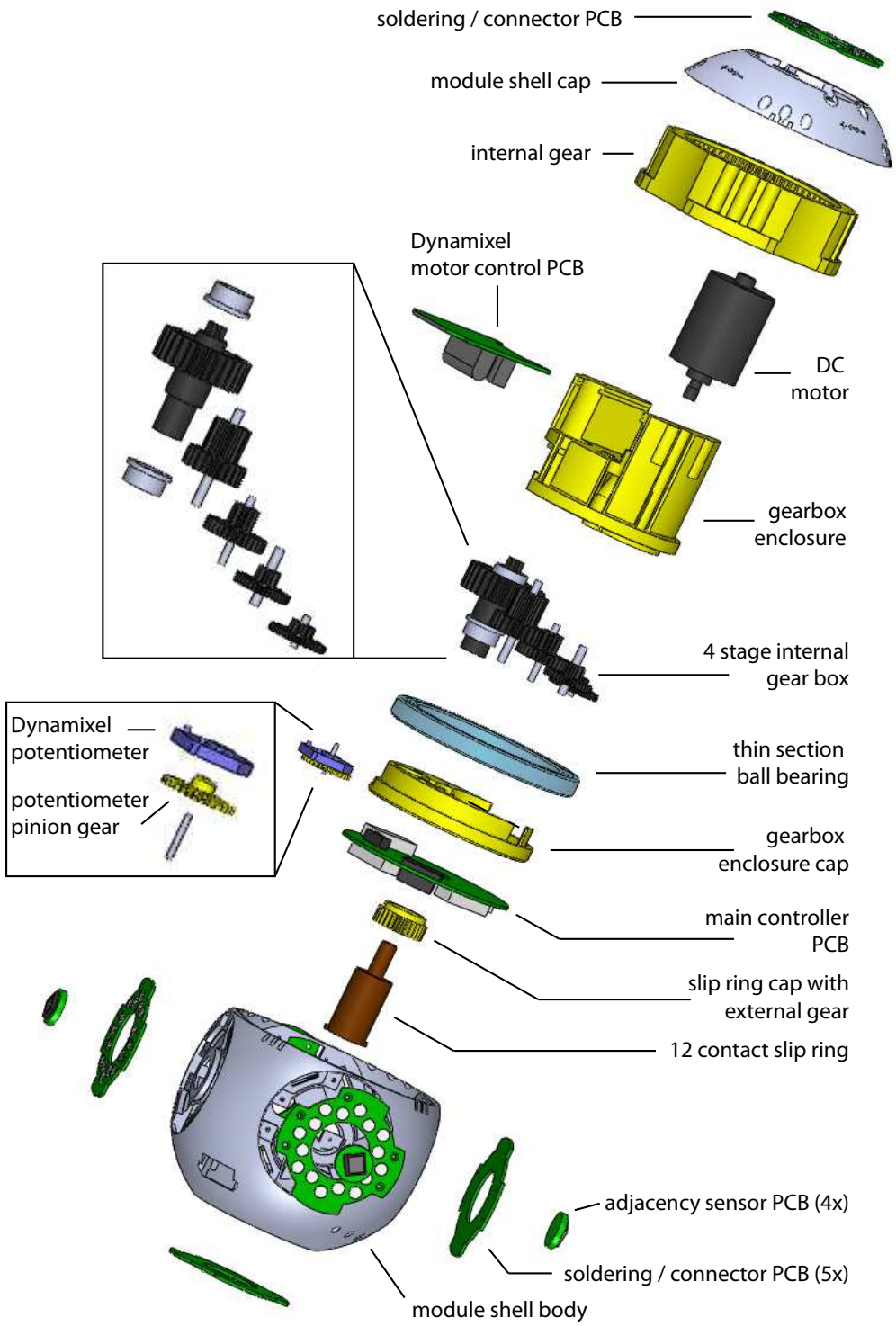


Figure 7.2: Exploded view of Soldercubes actuation module. Fasteners, magnets, and wires omitted for clarity.

many applications, its outer shape is not ideal for fitting into a cubic space. Therefore, the AX-12A internal components are rearranged and placed in a custom 3D-printed enclosure to make improved use of the cube shaped space inside a Soldercubes module. Firstly, the four gears of the AX-12 gearbox as well as the DC motor are rearranged to leave a free channel through the center of the gearbox for wire routing and on shaft is substituted where two gears who formerly shared their axis of rotation have been moved apart. Second, the final gear stage which is conjoined with the output shaft in the original design is converted into the second to final stage which engages with a 3D-printed internal gear mounted inside the module's shell body. Third, the voltage regulator of the AX-12A controller board is unsoldered and mounted mirrored from the opposite side of the PCB to reduce the overall thickness of the PCB with components. Finally, the potentiometer is removed from its location on the AX-12A controller PCB, combined with a custom 3D-printed spur gear and mounted to engage with the gear placed on the slip ring. The last step moves the potentiometer away from its conventional location on the axis of rotation freeing up space for wire routing through the center of the gearbox. The photographs in Figure 7.1 shows several aspects of this gearbox design.

As mentioned above, the gearbox design and moving the potentiometer off the axis of rotation allow for routing wires through the center of the gearbox as shown in Figure 7.1(d). Further, the top of the slip ring can be placed to reach into the gearbox, yielding an overall length reduction of approximately 10 mm in the gearbox dimensions. The resulting smaller overall dimensions of the module outer shell, higher density of modules, and reduced torque per connected module make this a worthwhile optimization. Figure 7.1(b) shows a top view of the gearbox assembly with the center opening for inserting the slip ring.

The load bearing interface between the two parts of the actuation module is provided

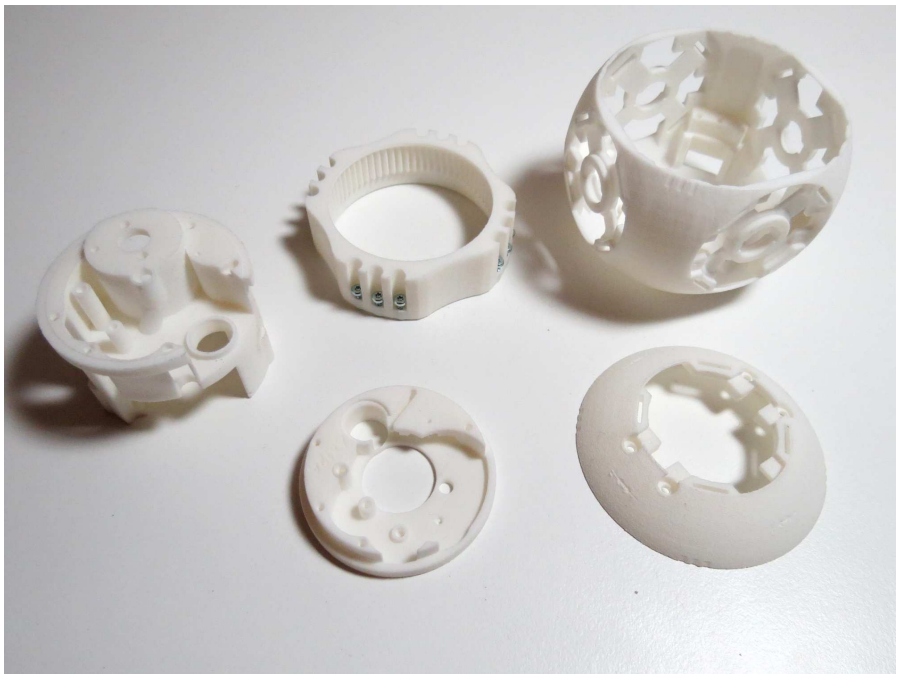


Figure 7.3: 3D-printed components of the actuation module. Top left to bottom right: Gearbox enclosure body, internal gear, module shell body, gearbox enclose cap, module shell cap.

by a four point contact thin section bearing with 1.5*in* bore. The gearbox is designed to fit inside this bearing and clamps it between its two parts as can be seen in Figure 7.1(c). The outer flange is similarly clamped between the cube shell and the large internal gear. This is part of the final reduction stage of the gearbox and can be seen during assembly in Figure 7.1(e).

Once fully assembled (Figure 7.1(f)), the outer dimensions of the actuation module are such that it fits a 55 mm cubic grid cell and the total weight is 120 g. Of comparable three-dimensional lattice type modular robot systems with actuated degrees of freedom, the Soldercubes module is the lightest and smallest. The efficient placement of components and integration of several functions into single components requires very finely detailed mechanical components. These components are 3D-printed in Objet Fullcure 720 material using an Objet 500 Connex polyjet resin printer with a resolution of

16 µm. Several other Objet materials as well as Selective Laser Sintering with Nylon powder (shown in Figure 7.3) were investigated for improving mechanical properties, but showed no benefit over the cheaper Fullcure 720 material.

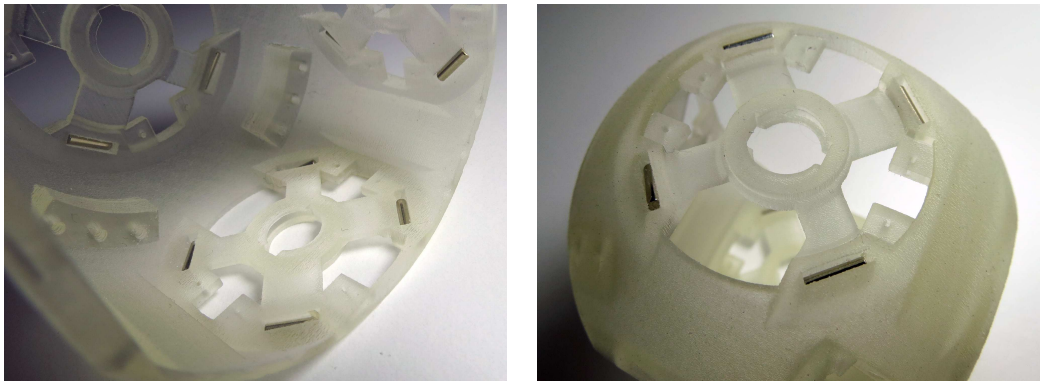
A complete bill of materials for the actuation module is provided in Appendix B, Table B.1. Figure 7.2 shows an exploded view of the actuation module.

7.2.3 Module Shell

Based on the design of the actuation module, the module shell is split into a larger part with space for five soldering connectors, and a smaller part with one soldering connector. The shape of the module which provides maximal internal space is a cube which would fill a lattice cell in the 3D grid in which the Soldercubes operates. However, because perfect cubes' vertices would reach into adjacent lattice cells during rotations, a competing requirement is to bring the module shape as close as possible to the insphere of the lattice cell in order for rotations to never interfere with neighboring cells. The design of the outer shell is further complicated by the constraint that the PCB which carries the soldering connector can only be manufactured and assembled flat and must be coincident with the boundary of the lattice cell.

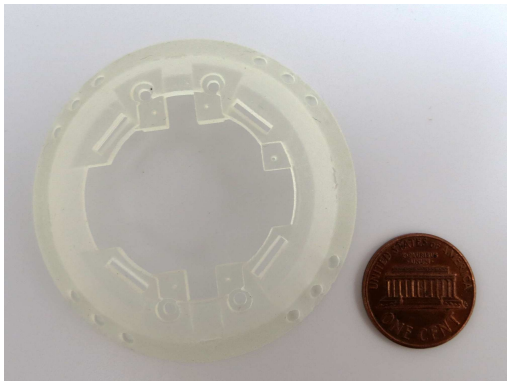
The shell design for Soldercubes is a tradeoff between those competing requirements: A cube with rounded corners taking up 75 % of the volume of the lattice cell. Each of the flat sides has a surface area of 821 mm^2, or 27.1 % of the lattice cell's side and is shared between the soldering connector PCB, an adjacency sensor, and alignment magnets .

Alignment magnets provide a virtual gradient for correct alignment during the ap-

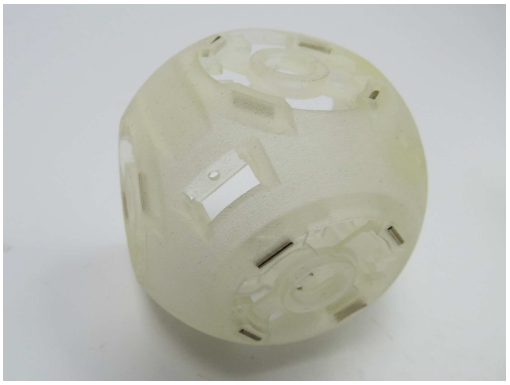


(a)

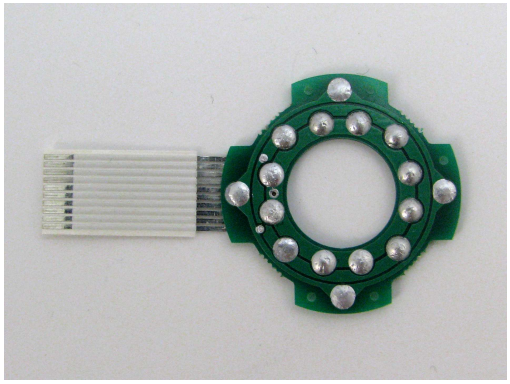
(b)



(c)



(d)



(e)

(f)

Figure 7.4: Photographs of Soldercube module shell components. (a) Inside view of module shell body. (b) Outside view of module shell body face. (c) Module cap (US penny for size reference). (d) Bottom view of shell body with opening for programming connector and holder for actuation module slip ring. (e) Soldering connector PCB with flat flex cable. (f) Miniature screws used for fastening connector PCBs into cube shell (US penny for size reference).

proach phase of a connection between two modules. While a mechanical gradient such as the curved cube surface described in Chapter 6 provides stronger guarantees that an incorrect connection is not formed, magnets have the benefits of forming a frictionless gradient and allowing for modules to slide past each other without getting blocked on a patterned module surface. Four $1/8$ in long $1/16$ in diameter Neodymium magnets are arranged equidistant at the circumference of the flat shell surface and provide sufficient force for individual modules to align when supported, for example on the substrate, but are not strong enough to interfere with robot actuation. The magnets rotate freely in a cavity that provides 0.3 mm radial clearance and has a small opening to the shell's exterior as well as a larger opening to the shell's interior from where the magnets can be snapped into position. Naturally, care must be taken that all magnets throughout the entire Soldercubes system have the same orientation. Figures 7.4(a) and 7.4(b) show the placement of alignment magnets in the cube shell from the inside and the outside of the shell respectively.

Holding the connector in place is challenging because no part of the module may extend beyond the contact plane of the soldering connector PCB but the assembly process necessitates that the connector PCBs is placed from the exterior of the shell, instead of inserting it into the module shell from inside the module. The connector PCBs were produced in a panel where each PCB is connected to a larger frame through small bridges that are broken to separate out the PCB. Because the remainders of those bridges are not filed off as one would normally do, they conveniently help hold the connector PCB in place in the cube shell during assembly. The connector PCB has five 1 mm diameter holes which align with small pilot holes in the shell. After creating a countersunk hole on top of the fabricated holes, flathead miniature screws with thread size 000-120 are screwed into the shell. Both $3/32in$ and $1/8in$ long screws, both shown on a penny for scale comparison in Figure 7.4(f), are used. While the screws are not designed to be

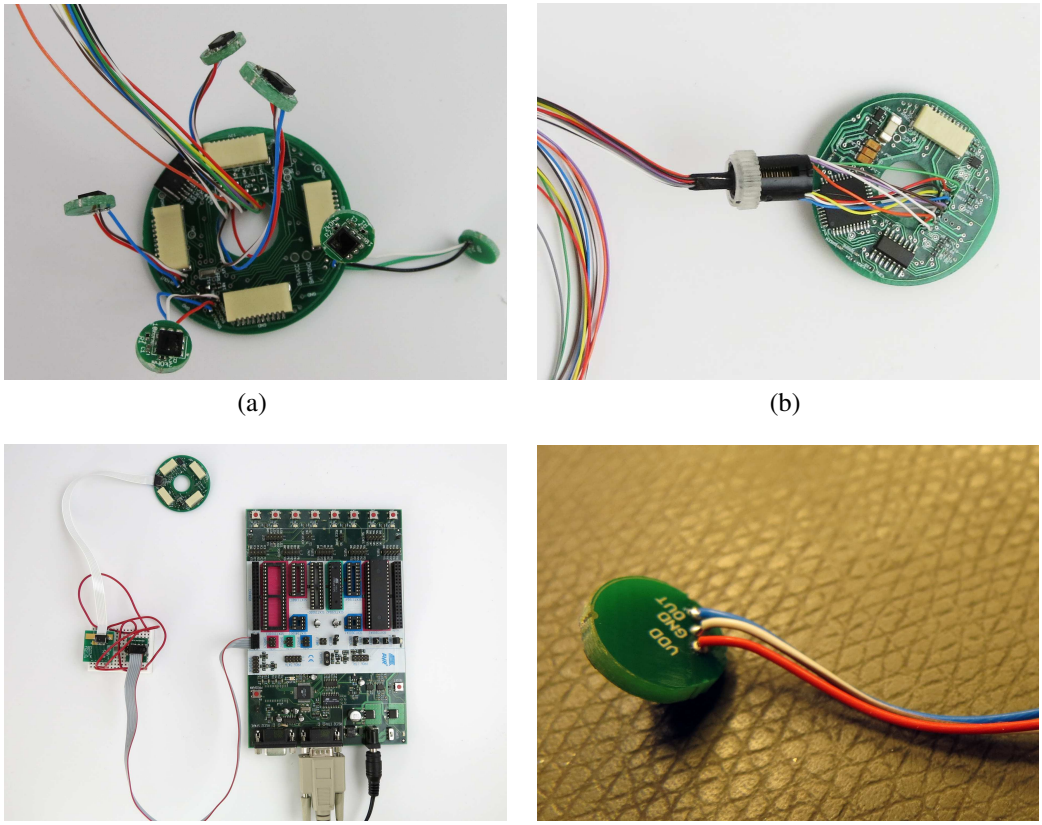
self-tapping, they act as such due to the softness of the 3D-printed Fullcure 720 material. To improve the retention force, the holes are pretreated with a fast curing low viscosity adhesive before inserting the screw.

While design for manufacturing considerations would suggest placing the adjacency sensor on the connector PCB, this is not possible because no part of the module may extend beyond the boundary of the lattice cell which is coincident with the outer surface of the connector PCB. Therefore, the adjacency sensor is implemented as a surface mount component on a small 8.5 mm diameter. To make assembly easy, the cavity in the center of each cube face is designed such that the PCB is inserted through a press fit and snaps into place inside the cavity. The additional slot through the cavity, best seen in Figure 7.4(d), allows for threading the adjacency sensor PCB through from inside the shell when it is already connected to the main controller PCB.

7.2.4 Electronic Components

Each Soldercube actuation module contains a total of 12 printed circuit boards. Ten are mounted in the module shell: Six soldering connector PCBs and four adjacency sensor PCBs. Inside the module a modified PCB extracted from the Dynamixel servo motor acts as motor controller. Finally, a main controller PCB connects all other PCBs.

All but two PCBs are on the same side of the rotational joint and the main controller PCB in the actuation module. One connector PCB and the motor controller PCB are on the other side and the slip ring connector described above in Section 7.2.2 provides the connection to the main controller PCB. The connection between adjacency sensor PCBs and the main controller PCB is permanent through three soldered down wires for supply as can be seen in Figure 7.5(a). Because the connector PCBs are inserted from



(a) (b) (c) (d)

Figure 7.5: Main controller PCB and adjacency sensor PCB. (a) Main controller PCB with six adjacency sensor breakout boards and 12 connection wires to the slip ring. (b) Main controller PCB with slip ring. (c) ISP programming the main PCB using a development board and custom adapter. (d) Adjacency sensor PCB with soldered connection wires.

the outside into the module shell, a permanent connection between the main controller PCB and the solder connector PCBs is not possible. Instead the connector PCB has a permanent connection to a 10-trace flat flexible cable, as shown in Figure 7.4(e), which is inserted into a low insertion force connector on the main controller board. Despite selecting the smallest footprint available, the connectors still are the most dominant component on the main controller board.

The adjacency sensor PCB contains only the Osram SFH 7741-Z proximity sensor

together with its essential peripherals. The range setting resistor is chosen for detection at 5 mm range to detect only directly adjacent modules but never modules one lattice cell away. If the proximity sensor were to operate continuously, a small but not negligible chance exists that two facing sensors detect each others emitters resulting in false readings. This risk is minimized by only providing power to the sensor when a reading is requested.

The connector PCBs are identical in design to those described in Chapter 3, and the development of the connector was in fact concurrent with the development of Soldercubes. To enable toggling the heater of the soldering connector the smallest available MOSFET device that can switch over 2 A was selected.

The main controller PCB is shown in Figure 7.6 as both photo and drawing. The core part of its functionality is the 8bit ATMEGA1284 microcontroller, which runs the control program controlling all functions of the module including switching connectors, sending motion commands, and communicating on the communication bus. The program can be updated using ISP programming both before the main controller PCB is inserted into the module shell, as shown in Figure 7.5(c), and after assembly by inserting the programming cable through the appropriate opening in the module shell, shown in Figure 7.4(d). An onboard resonator allows for operation at 16 MHz and two color LEDs directly connected to output pins provide low level debugging and status output.

The controller's two built-in UART ports are used for communication on the Soldercube system's global one-wire communication bus and for communication with the motor controller PCB, also through a one-wire serial bus. A quad bus buffer circuit is used to connect and disconnect the transmit and receive lines of both ports as appropriate.

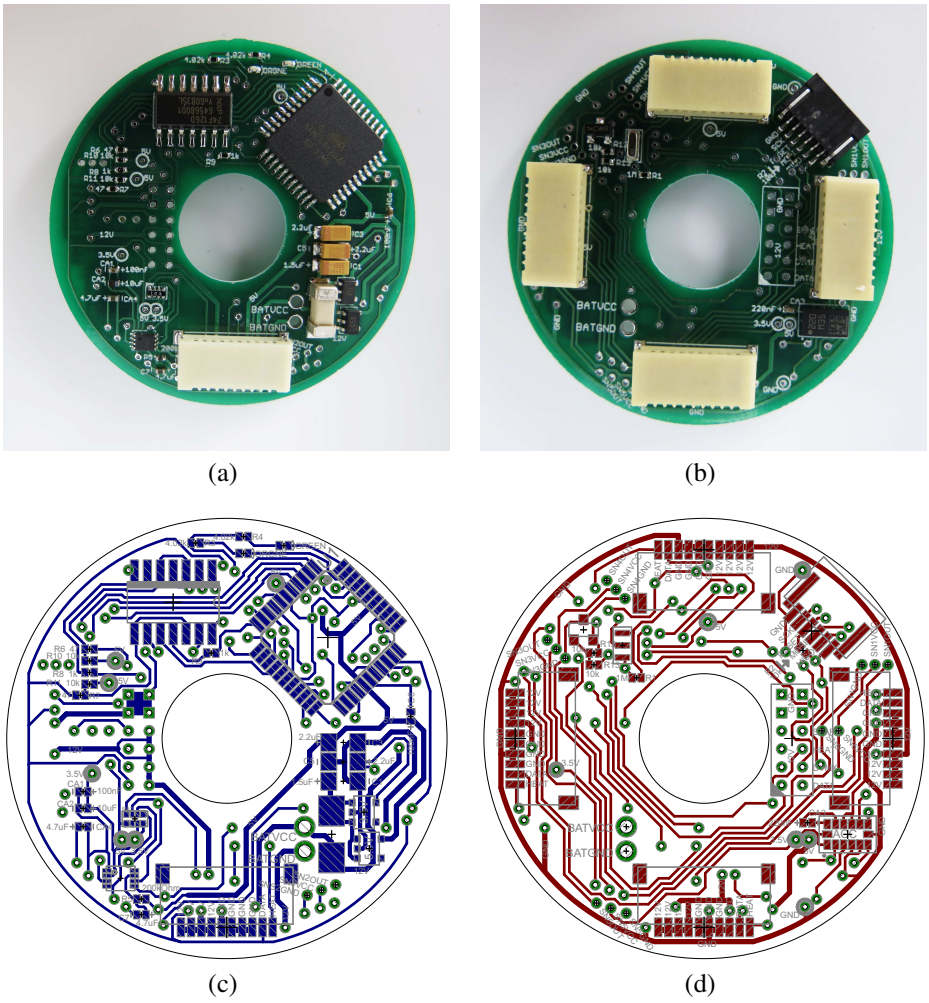


Figure 7.6: Photograph and schematic of main controller PCB. (a) Top view photograph of assembled PCB. (b) Bottom view photograph of assembled PCB. (c) Top side drawing of PCB. (d) Bottom side drawing of PCB.

To provide additional sensing capabilities, a combined three axis accelerometer and magnetometer device, LSM303DLHC by ST Microelectronics, is included on the main controller PCB. It is connected to the controller through the I2C bus and in addition to being available as sensor input is used as a noise source to seed the random number generator whose output is used to provide non-deterministic wait times in the communication protocol.

The main controller PCB operates at 5 V with the exception of the accelerometer-magnetometer device which operates at 3 V. The 5 V voltage level is also distributed to the adjacency sensors. The supply voltage of nominally 12 V is distributed directly, without any intermediate devices or voltage drop, to all connector PCB and the motor controller PCB.

With the exception of the connector PCBs all circuit boards for all manufactured Soldercubes were assembled manually using a toaster oven. Detailed bills of material for all PCBs are provided in Appendix B.

7.2.5 Structural Module

The structural Soldercubes module is different from the actuation module in the following points:

- The structural module's shell has space for six adjacency modules because the space and electrical connection constraints of the actuation module do not apply.
- Gearbox, bearing, slip ring, and internal gear are replaced by a one-piece rigid structural component.
- The twelve wires connected to the slip ring connector in the actuation module are connected directly to their destinations in the structural module.

The weight of the structural module is 76 g. A complete bill of materials for the actuation module is provided in Appendix B, Table B.2.

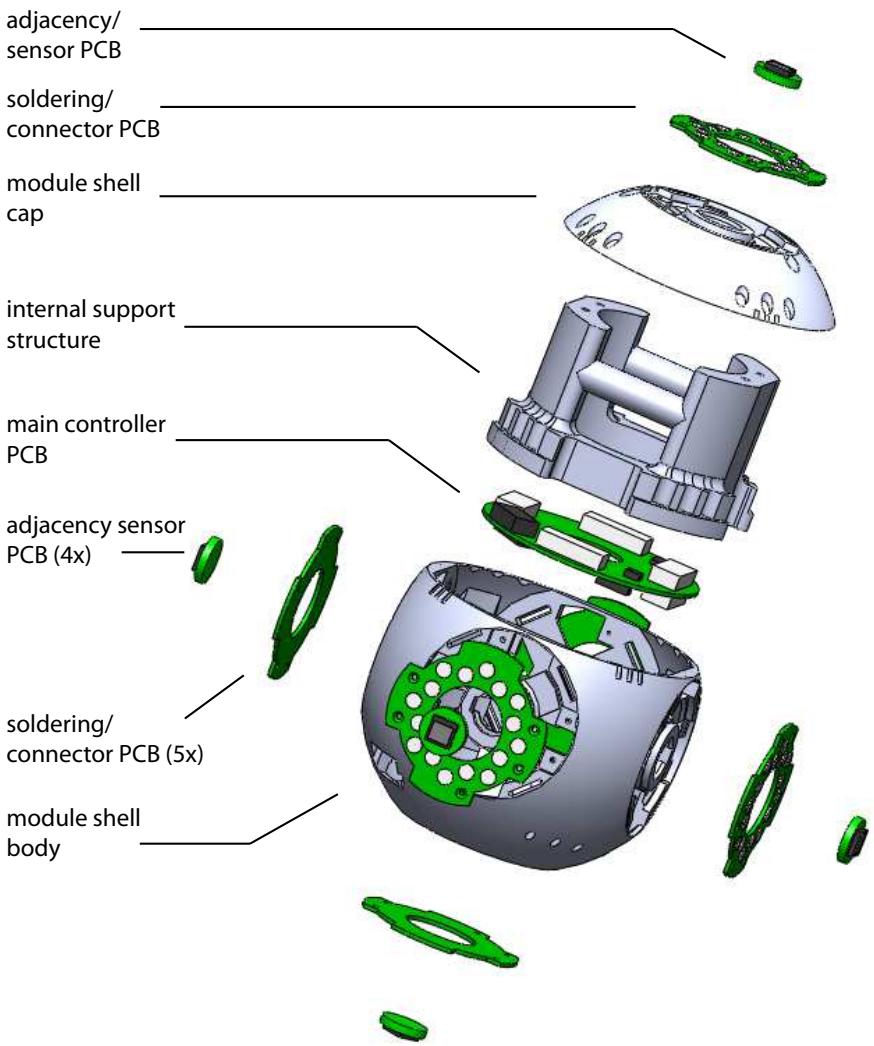


Figure 7.7: Exploded view of Soldercubes structural module. Fasteners, magnets, and wires omitted for clarity.

7.2.6 Energy Module

The design of the additional energy module type besides actuation and structural was driven both by conceptual fit and practical need: Conceptually it enables visual demonstration of the flow of energy through a network of interacting entities, a feature that is normally hidden when dealing with electrical energy transfer. The practical need for an energy modules arises because short interruptions of the power supply can occur when experimenting with large number of Soldercubes. Including batteries that can provide

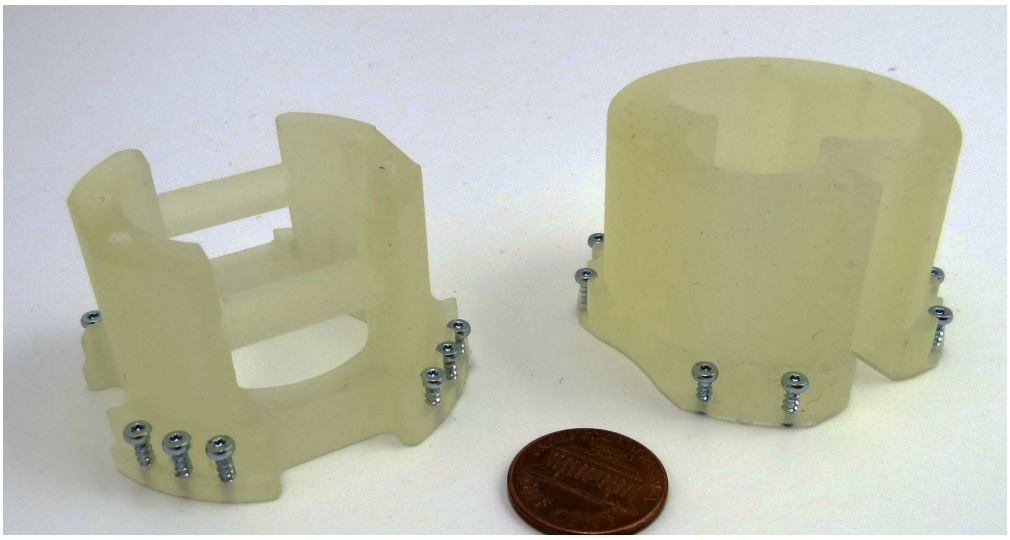
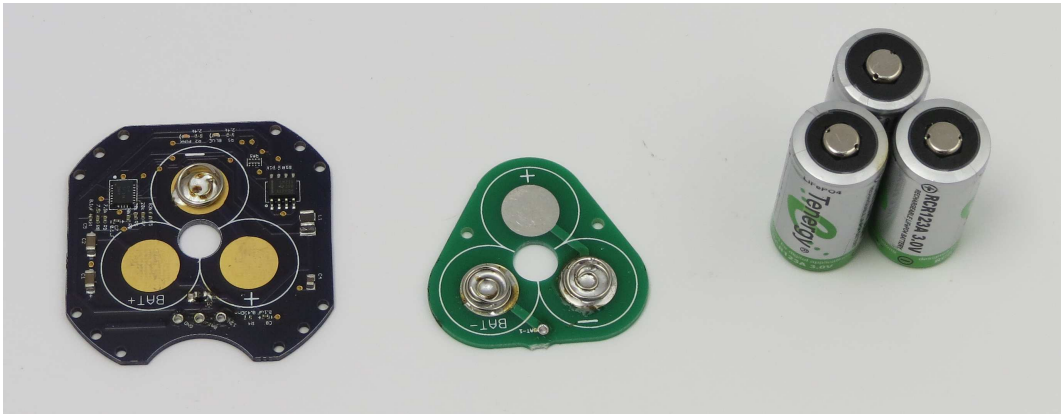


Figure 7.8: Internal structure of structural (left) and energy (right) module. US penny for scale reference.

power to modules in the Soldercube robot is a convenient way to mitigate this problem.

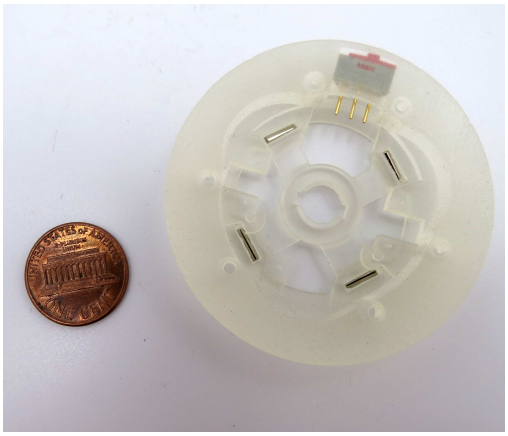
The choice of energy storage device for the energy module is limited by the specification of the previously designed actuation module: The device must provide a voltage around the system supply voltage of 12 V (with deviations of up to 3 V above and below tolerable), must provide at least 0.5 A current to be able to power one soldering connector at a time, and must fit inside a module shell. In order to make the energy module practically useful the power storage device should be either rechargeable or easily replaced.

The RCR123A series of lithium iron phosphate ($LiFePO_4$) rechargeable batteries marketed for consumer electronics meets all requirements stated above when three cells are connected in series. Marginal benefit of using this type of battery in Soldercubes are the thermal stability even when used incorrectly, and the easy to control characteristic charging curve. While the nominal voltage of each cell is 3.2 V it actually reaches up to 4.0 V when fully charged and supplies a nominal 0.5 A of current.



(a)

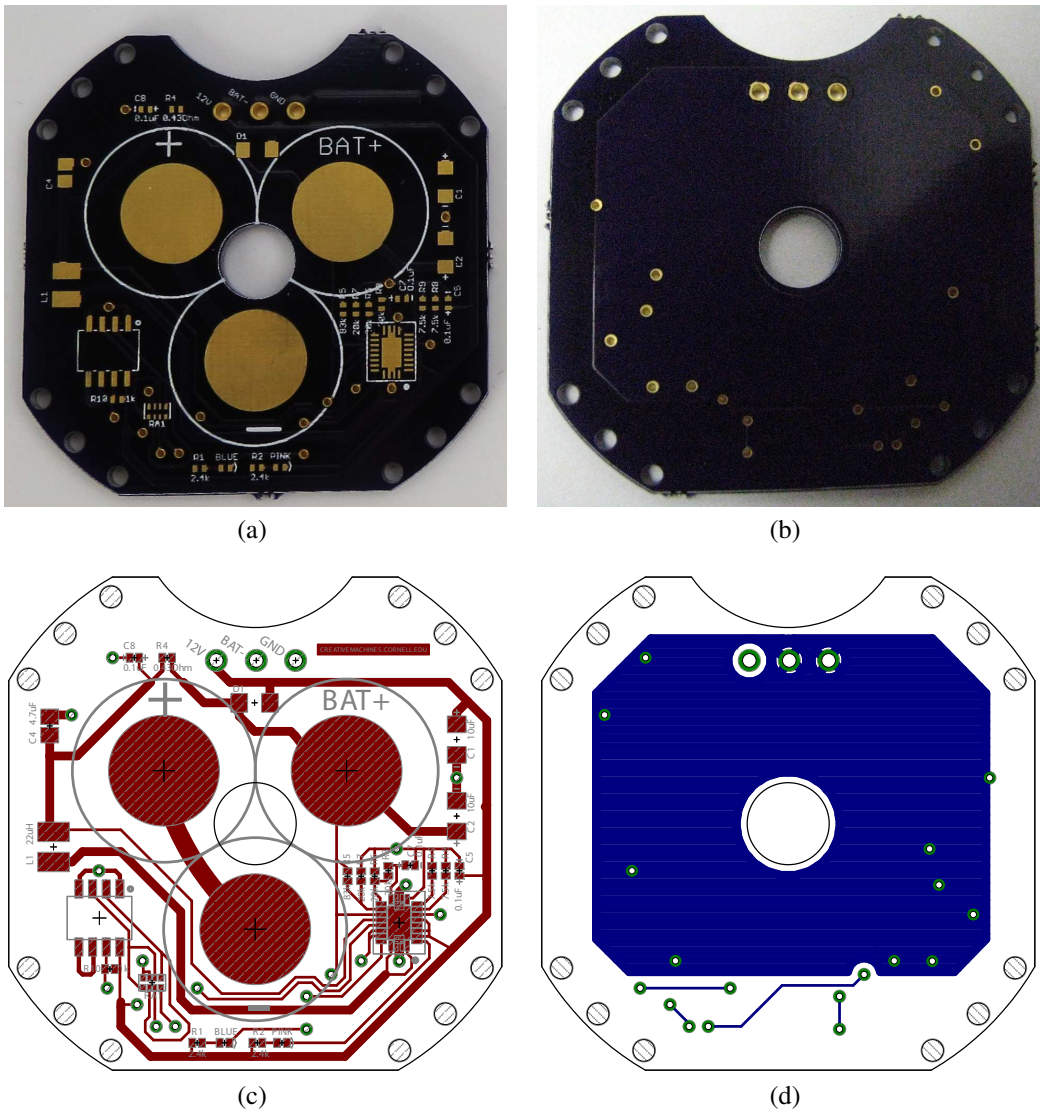
(b)



(c)

Figure 7.9: Components of the Soldercubes energy module. (a) Left: Battery controller PCB. Center: Battery holder with spring mounts. Right: Three RC123A $LiFePO_4$ rechargeable battery cells. (b) Test assembly of battery controller, holder and cells in the same arrangement as inside the energy module. (c) Specifically designed module shell cap with cavity for top battery holder PCB and externally accessible battery switch.

The Soldercube energy module contains a separate battery charger PCB, which controls the charging of the rechargeable battery cells and is mounted parallel to and on top of the main controller PCB. The design of the charge controller is complicated for three reasons: First, the energy module must seamlessly switch between the externally powered charging state and locally powered discharge state, similar to how most rechargeable consumer electronics products operate. Second, the same line that supplies power



(a) (b)

(c) (d)

Figure 7.10: Battery charge controller PCB. (a) Top view of unassembled PCB. (b) Bottom view of PCB. (c) Schematic drawing top. (d) Schematic drawing bottom.

while charging must be supplied with power during the discharge state in order to supply energy to any connected non-energy modules. And finally, multiple energy modules should be able to operate within one system without one energy module charging the other. The latter requirement is complicated further because multiple energy modules may not be charged to the same extend and will as a result supply different voltage levels.

The charge controller is designed around the Texas Instruments BQ24105 integrated circuit (IC) battery charge controller. At design time this IC was the only commercially available charge controller that can be configured to be compatible with multiple $LiFePO_4$ cells in series. The IC is configured to provide a 310 mA maximal charging current, and terminates the charge once the charging current drops below 30 mA. No charge termination hysteresis is applied to reduce the external component count which results in frequent switching between charging and not charging states around the time of reaching a full charge. The thermal protection function of the IC are not used.

The problem of only drawing current from the battery is solved by only charging the batteries to an in-series voltage of 10.8 V and placing a diode with a forward voltage drop of 0.5 V between the positive terminal of the series and the power supply line. This has the result that current is only drawn from the battery when the power supply line voltage drops below 10.3 V. It also means that when powered by energy modules, the power supply line voltage does not exceed 10.3 V. The forward voltage drop across the aforementioned diode is used to prevent energy modules from charging each other: The enable pin of the charge controller IC is only enabled when the power supply line voltage exceeds the voltage at the positive terminal of the battery series. When only one energy module is present within a system, the power supply line is always 0.5 V lower than the positive battery terminal and the charge controller IC will always be disabled. When multiple energy modules are present within a system, this setup prevents energy modules from charging each other as long as the voltage across the batteries does not vary more than 0.5 V.

Any mount for battery cells needs to ensure that there is continuous electrical contact between the mount and the battery cell terminals, which is commonly achieved through springs. The interior of the Soldercube module does not leave space for a separate bat-

tery mounting assembly which is why the mounting function is designed into the battery controller PCB. Figure 7.10 shows the battery charge controller PCB: All components are around around the perimeter leaving space in the center for three battery electrodes and a hole through which the connecting cables for the module's sixth adjacency sensor and soldering connector are routed. The polarity of the battery cells is such that one negative electrode touches the charge controller PCB; this electrode has a battery connector spring soldered to it.

The bottom of the battery charger board faces the main controller board and the tolerances of the cube shell are such that the components on the controller PCB might touch the battery charger PCB. To avoid unintended connections no components are placed at the bottom of the battery charger PCB and all necessary plated drill holes are constraint to locations that are not on top of tall components on the main controller PCB. Figure 7.11 shows an exploded view of the energy module design.

The counterpart to the mount integrated in the charger PCB is a passive triangular shape PCB on the other side of the battery cells that solely contains three appropriately connected electrodes. A custom module shell cap contains a matching cavity with two pins to hold the battery holder board in place and provide orientation. The shell cap is fastened onto the internal structure of the module, as is alsi the case for all other module types, which compresses the springs that touch the battery cell electrodes. In effect, the module shell provides the mechanical function normally provided by a dedicated battery holder.

The weight of the energy module, including batteries, is 138 g, making it only marginally heavier than the actuation module. A complete bill of materials for the energy module is provided in Appendix B, Table B.3.

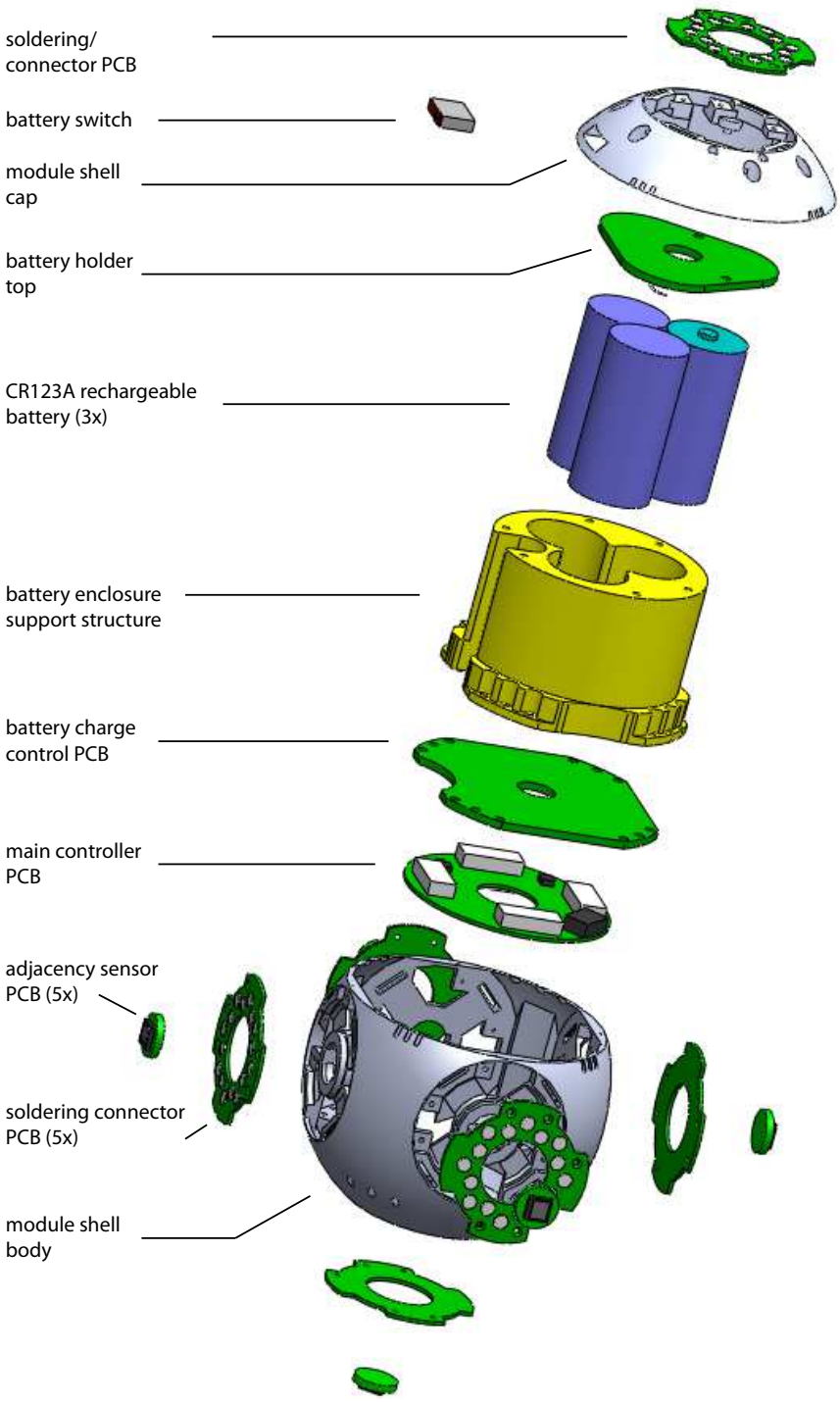
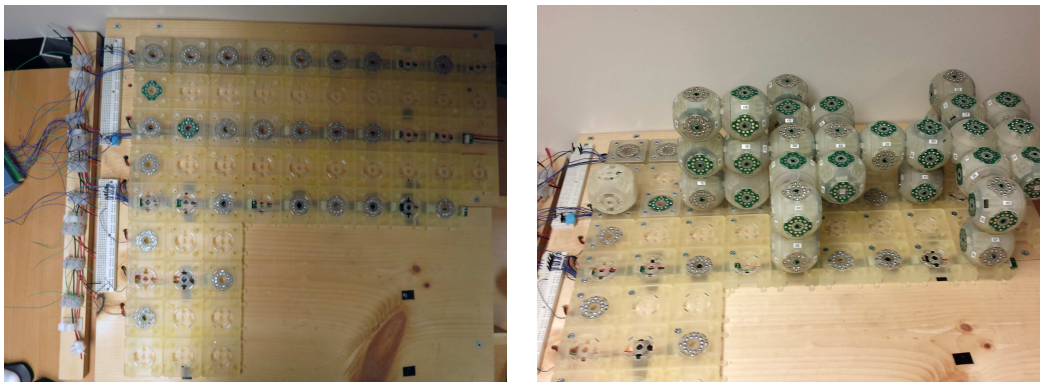


Figure 7.11: Exploded view of Soldercubes energy module. Fasteners, magnets, and wires omitted for clarity.



(a) (b)

Figure 7.12: Partially assembled ten by ten tile substrate: (a) Top view. (b) With a Soldercubes robot assembly.

7.2.7 Substrate

A substrate is necessary to enforce the constraint that all Soldercube modules in a system are aligned with a 3D grid when interacting. The substrate consists of tiles that are patterned at an interval of the same 55 mm length as the length of a lattice cell. Each substrate tile contains a soldering connector identical to those in the Soldercube modules making the substrate functionally equivalent to a an assembled layer of modules to which other modules can connect. Each substrate tile is connected to the shared power supply and ground lines to enable the soldering connector function and to transmit power to any attached module. In addition, the shared communication bus is connected to all substrate tiles in order to transmit the bus signal to any attached module. This has the effect that all Soldercubes directly or indirectly connected to the substrate form one system with one shared communication bus.

Every substrate tile's heater control signal is connected individually to a digital output peripheral connected to a PC from where it is controlled from a simple graphical user interface. This means that each substrate tile's soldering connector can form and break connections independently of all other substrate tiles and modules and does not

require the presence of a communication bus to do so.

For simplicity of implementation, the experiments described in the following sections used a partially assembled ten by ten substrate shown in 7.12 where only those tiles that could be connected to a module are outfitted with a connector PCB.

7.3 Control System

The control system of a Soldercube modular robot system consists of the control programs operating separately in each module, the communication system between modules, and the high level control that defines behaviors of the module assembly. The following sections describe the implementation of these components in the Soldercube system. The high level control is implemented for manual operation through a graphical user interface (GUI) on a connected PC, but future directions towards autonomous operation are suggested.

7.3.1 Embedded Module Controller

The embedded module controller program runs on the microcontroller housed on each modules main controller PCB which is described in Section 7.2.4. All module types run the same control program but the program flow varies depending on the module type flag stored in the microcontroller's EEPROM. The program is written in C and compiled and transmitted to the module using Atmel's free Atmel Studio software.

The primary function of the embedded program is to send and receive commands on the communication bus and interface with the twelve peripheral circuit boards as well as

the onboard accelerometer-magnetometer device. Each adjacency sensor is connected to a separate digital input pin with one digital output switching all sensors on and off together. Similarly, each soldering connector PCB is connected to one digital output pin to control heating on the connector. Communication with the motor controller is through a single wire serial serial bus. Because the communication with the motor controller is transaction based, each outgoing message results in zero or one incoming messages, and the communication is synchronous blocking all other module functions. Conversely, incoming messages can arrive on the Soldercubes communication bus at any time which is why the receipt of those is interrupt based. Whenever a byte arrives, it is placed in an incoming buffer. In the programs main loop, the incoming buffer is checked for new messages continuously and if a new message is found the appropriate action is taken immediately.

Several program functions rely on the presence of a local clock which is why time is counted using a timer interrupt at a millisecond resolution. One time dependent function is control of the soldering connectors: To avoid unintentionally heating connectors, for example after a system reset or due to human error, it is only possible to enable heaters with a finite timeout after which they automatically switch off. Whenever a heating command is received on the communications bus, the time of switching the heater on is stored in a variable to which the current time is compared at every iteration of the main control loop until the time-out is reached and the heater is automatically switched off. Several other features including timeouts in the motor controller communication, and allowing the adjacency sensors to initialize for a specified time also rely on local timekeeping.

Besides the normal operation mode the control program can be switched to a number of special purpose modes. In *Debug Mode* the module broadcasts human readable

ASCII messages on the communications bus, representing either potentiometer readings, accelerometer/magnetometer readings, or both. In *Calibration Mode* the program listens for a special subset of commands for rotating in arbitrary increments and storing the current position as setpoints representing 90° increments. During normal operation these setpoints will be used as target positions. Finally, *Potentiometer Alignment Mode* is used to turn the potentiometer to a valid position before inserting the gearbox assembly into the cube shell during the assembly of a module. Because the potentiometer has a 60° wide dead band in which it provides no or incorrect readings, it must be initially positioned such that none of the four equally spaced setpoints would fall inside this dead band. In order to achieve this, the green and red output LEDs on the main controller board are used to indicate whether four valid setpoints are achievable if the gearbox were to be mated with the module shell in such a way that it were currently aligned at one of the 90° increments. Once the potentiometer has been turned manually to such a position, it is up to the person assembling the module to align the gearbox approximately such that the edges of the motor controller PCB are parallel to the edges of the shell. When computing the validity of the current potentiometer reading, the software takes into account that it is impossible to achieve perfect orientation and only indicates positions at least 10° away from an invalid position as valid.

The aforementioned 60° interval over which the potentiometer does not return correct readings also needs to be addressed in software to produce correct motions between any two set points in both directions. The embedded Dynamixel firmware on the motor control PCB's microcontroller cannot be changed. It accepts commands to move to any positions in the 300° range covered by the potentiometer or can be set to rotate continuously. When commanded to move to a specific position, the firmware always choses the path that does not cross the potentiometer's dead band, even if this results in a longer rotation. To support 90°, 180°, and 270° rotations in either direction including those that

cross the region not covered by the potentiometer, the main controller program switches the motor controller into continuous rotation mode with rotation in the desired direction whenever any rotation command is received. It then proceeds to continuously poll the motor controller for the current potentiometer reading until this reading comes within a 10° range of the target location. At this point the motor controller is switched back to position control mode and instructed to move to and hold the target position. Motor parameters that define the proportional gain of the motor controller as well as maximum torque and deadband around the target location to avoid oscillation were carefully tuned to yield a smooth transition between continuous rotation mode and position mode in the motor controller. During the time when the motor controller is in continuos rotation mode and the main controller is polling for potentiometer readings, it is up to the main controller to detect wether the current potentiometer reading falls within the dead band. This is achieved using multiple metrics: First it is detected when the potentiometer approaches and reaches either limit upon which the controller assumes it is in the dead band until the other limit is seen and departed from. Because the potentiometer sends non-limit readings while transitioning through the dead band, a numerical derivative is taken at each reading. If the derivative falls within the range of normal motion for several subsequent readings any assumption about being in the dead band is discarded. Conversely, when large derivates are detected for several subsequent readings, it is assumed that the current readings stem from the dead band. Even with this two layered detection system in place, approximately 20 % of potentiometer still showed unexpected readings while transitioning through the dead band and were discarded.

7.3.2 Communication System

Section 5.4 on communication in modular robots describes, in part, the serial bus communication system of Soldercubes: The microcontrollers in each module communicate through their UART ports whose send and receive pins are connected through a buffer circuit with a one-wire bus. The $0x21$ (ASCII character "!") address is reserved as broadcast address to send commands to all modules in a Soldercubes system.

Because in current applications only the human controlled base station sends command messages, no special provisions to avoid multiple modules sending messages at the same time have been taken. The exception to this is at startup, when each cube broadcasts a "/" message to report its presence. The probability of collisions at this point are avoided by waiting for a random amount of time with the random timer seeded from the noise in accelerometer readings. In addition, modules do check for incoming data before enabling the transmit direction in the bus buffer, but it is still possible for one cube to start sending data during the time interval it takes another module to switch from receive to transmit mode. For applications where each module could transmit data independently a more sophisticated carrier sense multiple access (CSMA) scheme could be implemented without hardware changes.

7.3.3 Synchronized Motions

When interacting with robots that contain more than one actuated module a facility for synchronizing the clocks of multiple modules might be required. For example, any robot able to locomote including those described below in Sections 7.5.3 and 7.5.4, requires two actuation modules to operate in a synchronized fashion. The Soldercubes communication protocol enables two ways of achieving this.

A first method to achieve synchronized motion first broadcasts the "T" command to reset the clock of all modules in the system[3]. As a second step, "R" commands that trigger a rotation are sent with the fourth data byte set to a non-zero value which specifies the absolute clock time at which the rotation should start. Upon receipt of this "R" message the module immediately goes into a blocking wait mode resulting in the start of actuation at the precise (to the millisecond) clock time requested.

A second method does not rely on resetting the clock ("T" command) and instead sends the "R" command immediately with the fourth data byte set to *0xFF* indicating that this command should be buffered. A second broadcast "R" command (to the *0x21* broadcast address) triggers the buffered rotation commands on all modules that previously received a buffered rotation command. The benefit of this second method is that it is not blocking and the controller remains in its main program loop during the intermediate wait time. This means that other intermediate commands can be sent to the module and that the first "R" message will be acknowledged with an "r" message by the module. The latter is most useful when the communication bus is unreliable, because the initiator of the commands receives a confirmation of receipt from all participating modules before triggering the physical motion. The disadvantage of this second method over the first is that any intermediate command received by a participating module might result in a delay of this module's start of motion.

7.3.4 Graphical User Interface

The ultimate goal of building any robotic system is a large degree of autonomy with no or only high level inputs from users. During the development of the system, how-

[3]The implementation of "T" command for clock resets is such that all other functions relying on the clock are not affected, for example countdown timers for heating soldering connectors still time out after the requested time interval.

ever, low level access to many functions is required. The control software developed for the Soldercubes system has the goal to present a graphical user interface (GUI) to control low level system functions during development, and easily scale into an intuitive interface for triggering more complex, high level behaviors.

Figures 7.13 and 7.14 show the two main screens of the control software. The screen in Figure 7.13 allows for connections to the Soldercubes serial bus to be created through a serial port or the electricImp internet bridge (see Section 7.7.1). Arbitrary messages can be composed and sent individually. Both a raw and a parsed view on all bus communications is available in this view. A separate tab (Figure 7.14) can be opened for each module showing graphical interfaces to all common control and debugging functions on a per module level. For actuation modules this view includes calibration, actuation, and sensor functions, as well as direct access to all Dynamixel servo motor parameters. The developer console available from the software development framework used to create the GUI gives convenient command line access to all features and allows for arbitrary scripts to be written in Javascript.

The basic interaction of the control software with the system is through sending and receiving individual messages. Interactions that require multiple messages to be sent or received are easily programmed and exposed to the GUI as *Macros*. This feature becomes quickly necessary when interacting with moving robots that require synchronization as described above in Section 7.3.3. For example, for the demonstration of a four legged robot in Section 7.5.4 each step requires eight instruction messages to be sent, their respective acknowledgements received, and a final broadcast message for the motion to be triggered. If any of the initial instructions is not acknowledge through an appropriate message, the step must not be executed to avoid incomplete movements that might result in damage to the robot. This logic was wrapped in a macro function trig-

gered by one GUI button. It is easy to imagine how even more complex commands can be created by adding additional layers of abstraction beyond this simple macro, resulting in reactive and increasingly autonomous robot behavior.

The control software is packaged as a Google Chrome App [180]. This allows for easy distribution without an installer to any computer on any operating system that runs the Google Chrome web browser. Google Chrome Apps are written using web standards including HTML and JavaScript with custom extensions for accessing systems functions, such as the serial port for the Soldercubes control software. The implementation of the control software uses the AngularJS framework for web applications in Javascript [181]. All unique components such as serial port access and parsing of the data stream are exposed as services that are accessed by controllers, which implement user facing features that are then exposed graphically through HTML templates. The visual elements of the GUI use the Twitter Bootstrap library [182].

7.4 Robot Ecology Simulator

In order to plan and visualize experiments with Soldercubes offline a simulator for robot kinematics was developed. The simulator is implemented in HTML and Javascript and can be navigated to as a normal website. The interactive 3D visualization uses the open-source *three.js* Javascript library. Using Javascript or simple mouse interactions in the GUI, modules can be placed and connections between modules can be defined. Robots, clusters of connected modules, are automatically detected. Motion commands can be defined in Javascript code or through mouse interactions in the GUI and the resulting motion of the kinematic chain of modules is computed and displayed. The simulator does not simulate dynamics or friction but does check for collisions between modules.

Figure 7.13: Screenshot of Soldercubes control GUI main screen.

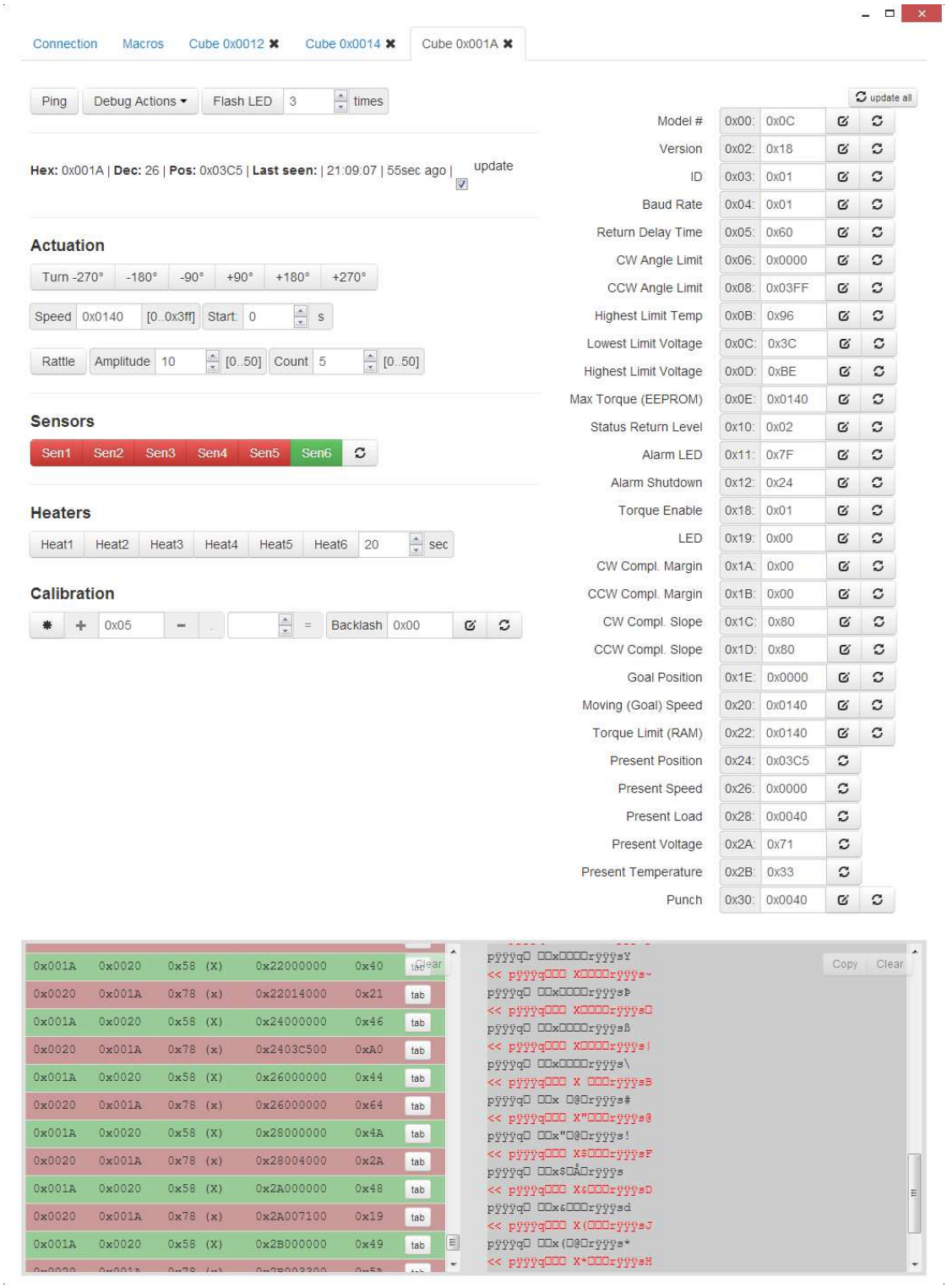


Figure 7.14: Screenshot of Soldercubes control GUI module control screen.

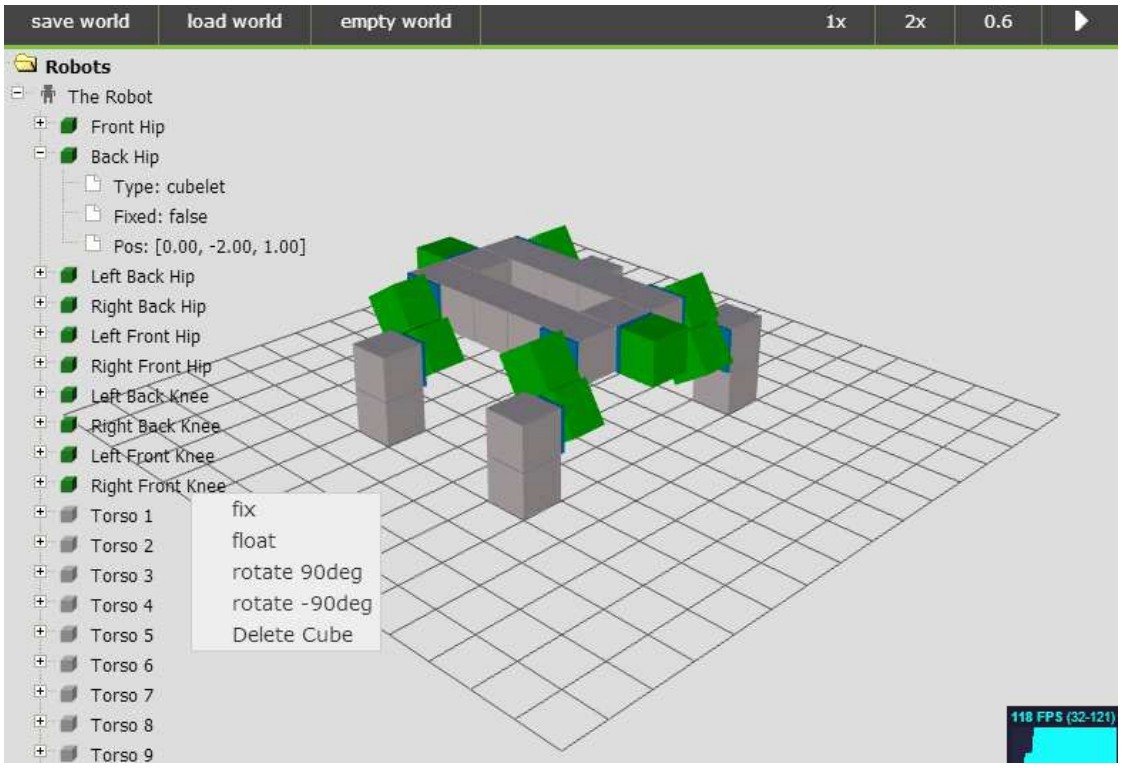


Figure 7.15: Screenshot of robot ecology simulator web application. Central element is a 3D view of the simulated robot which can be interacted with using mouse controls similar to those found in 3D games. The controls in the top left allow for importing and exporting saved scenarios, the controls in the top right play any pre-scripted actions. The menu tree at the left gives access to all modules, robots, and actions.

A screenshot of the simulation of a four legged walking robot is shown in Figure 7.15.

Additional module types can easily be defined by extending the basic module class. This has proven instrumental in evaluation the placement of the actuator axis as discussed in Section 7.2.1. In addition to the Soldercubes, other module morphologies have been implemented including the Molecubes system whose self-replication demonstration was replicated in simulation.

In addition to giving visual feedback to aid users the simulator can be operated in "headless mode" without graphical output. This is useful for automatically evaluating

metrics of pre-generated scenario files, for example the distance travelled. Appendix A covers one application of this feature.

7.5 Experiments

7.5.1 Basic Pair

A pair of an actuation module and an energy module is enough to demonstrate the basic components of an autonomous robot: sensing, control, and actuation. For this experiment an actuation module with its axis of rotation oriented vertically is attached on top of an energy module. With the energy module resting on the ground, four of its adjacency sensors are visible. The software of the actuation module is unmodified from the standard described above in Section 7.3.1, while the embedded software of the energy module is modified to add the following behavior:

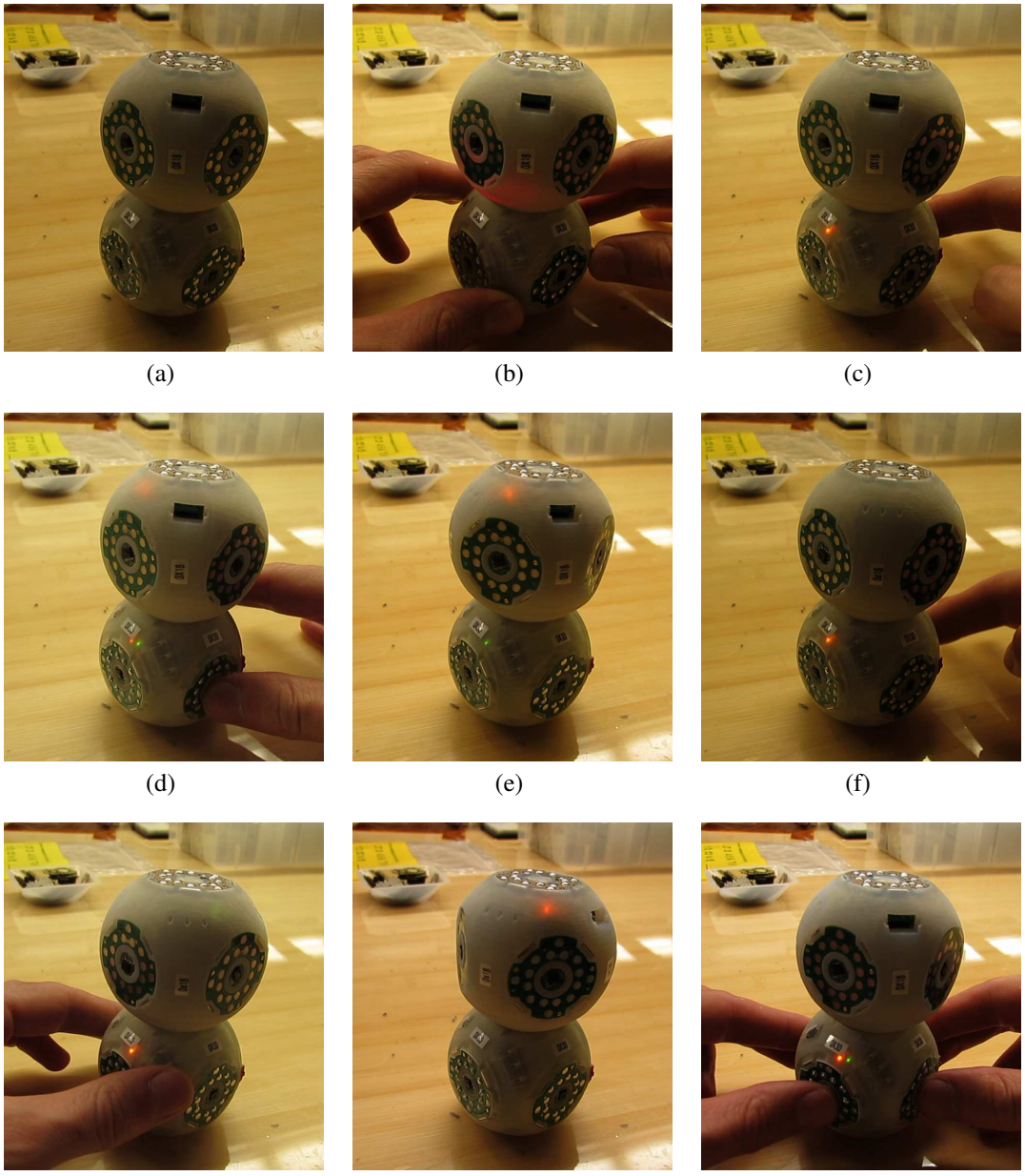
- When the north[4] facing adjacency sensor is obstructed, *command mode* is activated for four seconds and the LED is steady red.
- When the *east* facing adjacency sensor is obstructed while in command mode, a rotation command is sent to the attached actuation module to trigger a 90° *clockwise* rotation.
- When the *west* facing adjacency sensor is obstructed while in command mode, a rotation command is sent to the attached actuation module to trigger a 90° *counterclockwise* rotation.

[4]In this section, compass directions are used to describe relative orientation.

- When the north, east, south, and west facing adjacency sensors are obstructed, the top facing connector is heated in order to attach or release the actuation module.

The interaction with the basic pair assembly is shown in Figure 7.16 as a sequence of photographs. The following is a description of the steps performed in the experiment:

1. Figure 7.16(a): Energy module (bottom) and actuation module (top) attached an without power.
2. Figure 7.16(b): The energy module is switched on and supplies power to itself and the attached actuation module.
3. Figure 7.16(c): Touching the far adjacency sensor on the energy module triggers command mode indicated by a steady red light.
4. Figure 7.16(d): Touching the right adjacency sensor during command mode triggers the energy module to send a rotation instruction to the actuation module.
5. Figure 7.16(e): The actuation module rotates 90° counter-clockwise.
6. Figure 7.16(f): Command mode is entered again through touching the far adjacency sensor.
7. Figure 7.16(g): Another rotation instruction is triggered by touching the left adjacency sensor.
8. Figure 7.16(h): The actuation module rotates 90° clockwise.
9. Figure 7.16(i): Touching all four exposed adjacency sensors of the energy module at the same time triggers the connector to heat as indicated by both yellow and green lights being steadily on.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

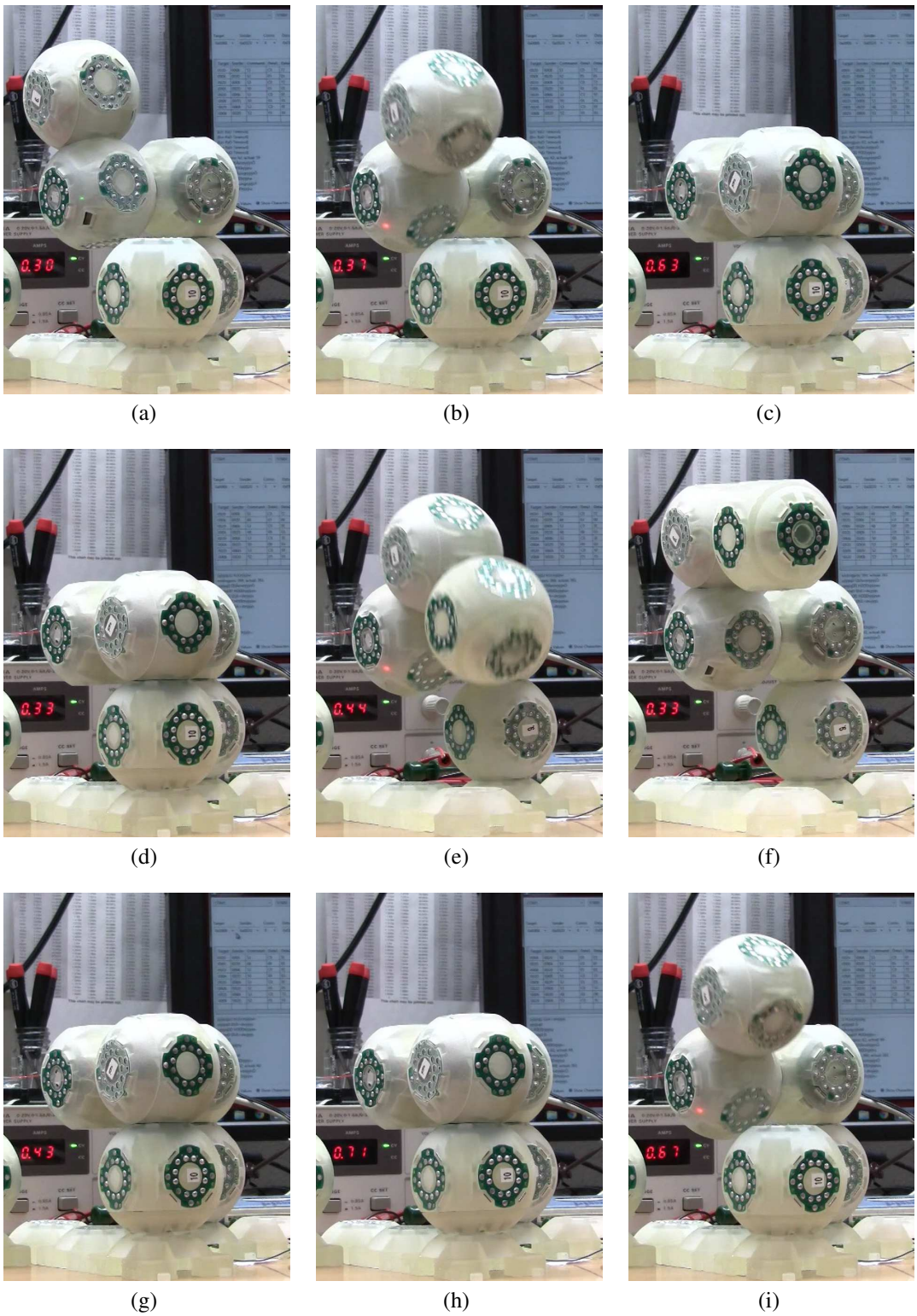
Figure 7.16: Sequence of photographs of the "basic pair" experiment. An energy module with modified control software (bottom) is connected to an actuation module (top). When the north facing adjacency sensor of the energy module is obstructed, the module switches in command mode awaiting further interactions while showing a steady green light. Depending on which adjacency sensor is obstructed next, the actuation module turns its actuated face by 90° in either direction.

In addition to demonstrating actuation, control, and sensing, this experiment also acts as a proof of concept for energy storage and distribution, and communication between modules – covering all basic functions required from a self-reconfiguring modular robot system in order to exhibit robot ecology behavior.

7.5.2 Module Acquisition

The acquisition and rejection of modules without external manipulation is a key components of realizing behaviors that include interactions with the environment. This simple experiment was devised as a basic demonstration of the capability of Soldercube modules to form and break connections between modules autonomously, as well as the ability to incorporate modules acquired from the environment into an existing robot. In Figure 7.17 a four module robot (back) picks up an additional module (front). The sequence can be broken down into four steps:

1. Figure 7.17(a) to 7.17(b): The arm of the four module robot descends on the single module.
2. Figure 7.17(c): The Soldercube connection is formed by heating the low melting point alloy as visible by an increase in the current consumption by approximately 500 mA.
3. Figure 7.17(d) to 7.17(f): The heaters are turned off (as visible by the drop in the current consumption) and the new module is lifted.
4. Figure 7.17(g) to 7.17(i): To release the module this sequence is reversed. The arm lowers the module to the substrate and breaks the connection by heating the low melting point alloy, allowing the arm to lift up to its starting position.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7.17: Demonstration of simple stationary robot picking up and later returning individual module.

The repeatability experiments described in Chapter 3 of this thesis may be a considered a continuation of this experiment which simulates the repeated connection and disconnection of two adjacent modules under controlled conditions.

7.5.3 Simple Walker

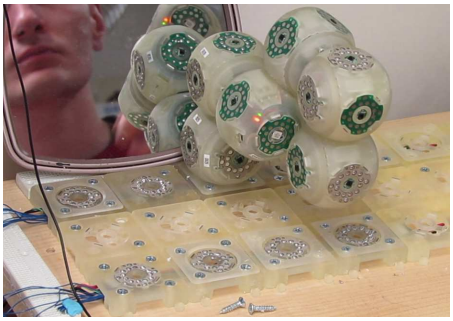
For any meaningful interaction between a robot and its environment, the robot must be able to move within its environment. Like for all other modular robot systems with one degree of freedom per module, the Soldercubes system requires at least two actuation modules within a robot for locomotion to be possible. Figure 7.18 shows a "simple walker" taking three consecutive steps on the substrate. The simple walker is a two legged robot consisting of two actuation modules and four structural modules. Walking over the substrate can be achieved by repeatedly performing the following sequence of interactions (yielding two steps):

1. Create a soldered bond between the right lower structural module and the substrate.
2. Actuate the left actuation module to turn 180° in the positive direction, and the left actuation module to turn the same amount in the negative direction resulting in the right half of the robot to make one step forward.
3. Create a soldered bond between the left lower structural module and the substrate.
4. Heat the soldered bond between the right lower structural module and the substrate in preparation for disconnecting.
5. Actuate both actuation modules as in step two to result in the right half of the robot disconnecting from the substrate and making one step forward.

(a)

(b)

(c)



(d)

(e)

(f)

(g)

Figure 7.18: Sequence of photographs of the "simple walker" experiment. A robot consisting of two actuation modules and four structural modules moves over the substrate.

The sequence required manual intervention for alignment of modules with the substrate. In addition to demonstrating the ability of Soldercube assemblies to locomote, the demonstration shown in Figure 7.18 confirms that the soldering connection method withstands sufficiently large forces to allow for cantilevered loads and realistic operation of module assemblies.

7.5.4 Acquisition of a New Function

After the previous sections described experiments which illustrate individual functions required to demonstrate an ecosystem of robots, the following experiment integrates all of them: A four legged robot capable of locomotion in one direction acquires modules from smaller autonomously moving robots in order to gain the capability of sideways movement. The smaller robot, which the four-legged feeds from, is identical to simple walker in Section 7.5.3.

First, the two-legged robot moves to the center of the substrate (Figure 7.19(a) to 7.19(d)). Next, the four-legged approaches the smaller robot by actuating such that its body moves forward (Figure 7.19(e)). To support the subsequent soldering connection a support block was manually placed under the four-legged's center head cube as best seen in Figure 7.19(d). When the four-legged and two-legged robots are aligned, the four-legged heats its front soldering connector PCB to melt the low melting point alloy and acquire the two-legged module (Figure 7.19(f)). After the connection has been formed the four-legged robot controls its new addition to undergo a series of transformations to create a fifth leg, enabling side-ways locomotion (Figure 7.19(g) to 7.20(g)). For this transformation it is also necessary to break the connection to one module (formerly the right foot of the two-legged) as seen in Figures 7.20(a) and 7.20(b). Some of the

re-configuration motions were manually supported, because lifting the weight of five modules through a single cantilevered connection is not supported due to insufficient mechanical design of the module shell (7.20(c)).

Figure 7.21 shows several thermal imaging photographs taken during the execution of the motion sequence in this experiment. These photographs show the heat generated by the electronic components in the module during normal operation which result in temperatures of up to 45 °C. While there certainly is room for improved thermal design, this is an expected temperature increase during continuous operation of any electromechanical system and validates the choice of a metal with a melting point above 60 °C for the soldering connector.

7.6 Simulated Experiments

Using the simulator described in Section 7.4 several more complex scenarios were planned. Each scenario is theoretically possible with the existing capabilities of Soldercubes and, once implemented, will demonstrate more complex interactions in the robot ecosystem. However, improvements in the mechanical design and reliability of the actuators are necessary to make these demonstration possible.

7.6.1 Module Collection and Growth

In this simulated experiment a four-legged walker of a similar morphology as the one seen in the demonstration in Section 7.5.4 is placed on a substrate with a large number of scattered modules. Through its morphology the robot is constrained to move on a line on which a two module tall obstacle is located.

(a)

(b)

(c)

(d)

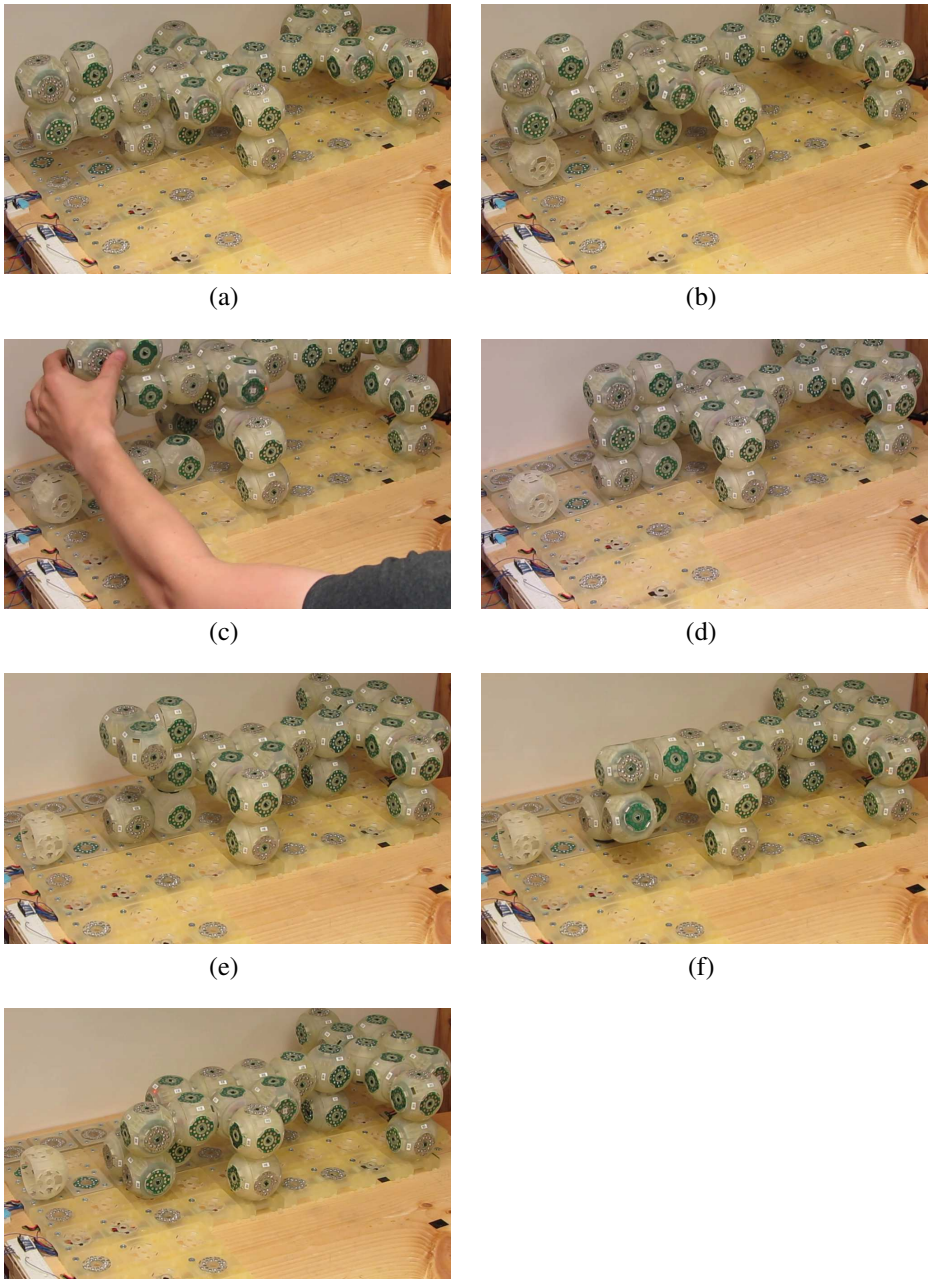
(e)

(f)

(g)

(h)

Figure 7.19: Sequence of photographs of the "feature acquisition" experiment. Part 1. A four-legged robot acquired an additional leg for sideways movement from a two-legged robot encountered in the environment.



(a) (b) (c) (d) (e) (f) (g)

Figure 7.20: Sequence of photographs of the “feature acquisition” experiment. Part 2.

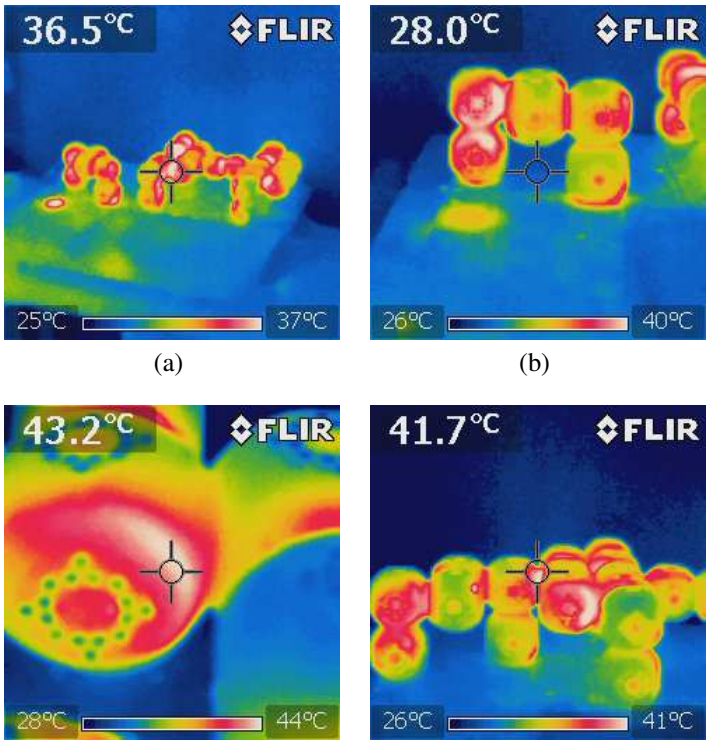


Figure 7.21: Thermal imaging photographs taken during the operation of the "Acquisition of a new function" experiment. No heater is active at the time the photos were taken. Heat generated during normal operation of the modules results in temperatures of up to 45 °C.

The robots behavior is such that it moves forwards and back within the bounds of the obstacle and the substrate dimensions until it finds suitable modules to grow additional front and rear legs. In addition to gathering modules that happen to be in a suitable location, the robot also "picks" and "drops" modules to place them in locations that are suitable for using them towards the objective of growing additional legs. Figure 7.22(g) shows the robot once the growth of new legs from modules acquired from the environment is completed. Because the newly grown legs are oriented at 90° to the original legs, the robot has acquired the capability to move sideways on the substrate (Figure 7.22(g) to 7.22(h) and Figure 7.22(i) to 7.22(k)). Through the simple interaction with the environment and a simple behavior of gathering suitable modules from the environment,

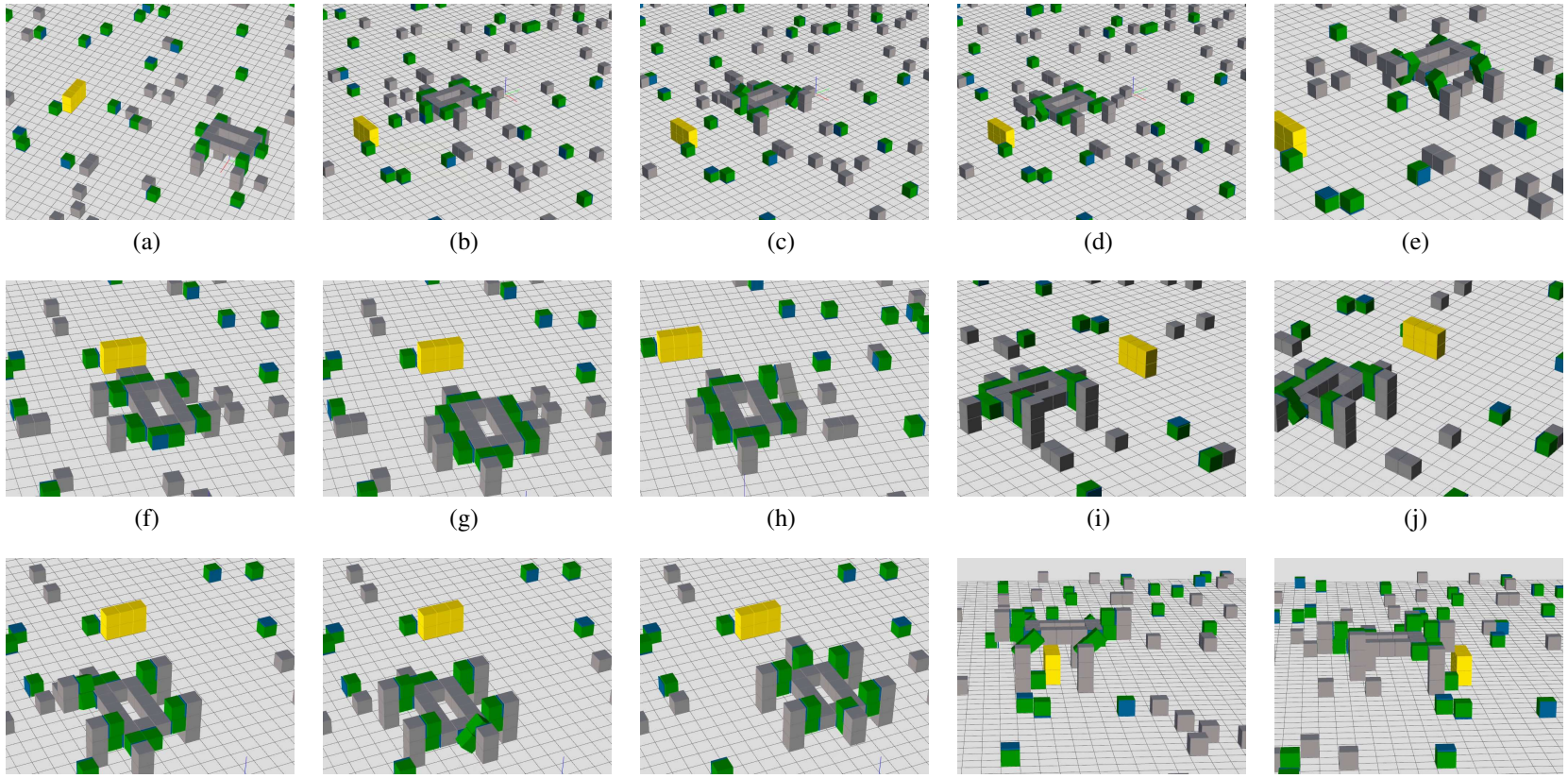


Figure 7.22: Sequence of photographs of the "basic pair" experiment. An energy module with modified control software (bottom) is connected to an actuation module (top). When the north facing adjacency sensor of the energy module is obstructed, the module switches in command mode awaiting further interactions while showing a steady green light. Depending on which adjacency sensor is obstructed next, the actuation module turns its actuated face by 0° in either direction.

the robot has removed the constraint of moving only on a line and could now reach any location on a flat 2D substrate.

A practical use of the newly grown capability is demonstrated when the robot moves sideways to acquire even more modules and connect them to the bottom of its original four legs, effectively growing taller in the process (Figure 7.22(i) and Figure 7.22(k)). Once four suitable modules have been found and harvested, the robot crosses a tall obstacle, an action that was not possible with the original legs (Figure 7.22(l) to 7.22(o)).

The growth of new abilities through the acquisition of resources from the environment is arguably a demonstration of the most high level properties of metabolism: The acquisition of resources that are useful, the modification of resources into more useful resources, and the rejection of modules that are not useful. These behaviors can be considered equivalent to biological organisms harvesting for food, metabolizing it, and emit waste products into the environment.

7.6.2 Predatory Behavior

The "predatory behavior" experiment is a more complete version of the feature acquisition behavior implemented with physical modules in Section 7.5.4 above. While the physical implementation only allowed for the acquisition of one new leg from another robot, this simulation continues to show the addition of a second rear leg (Figure 7.23(l)). With both new legs in place the robot is now able to move orthogonal to the direction it was previously constrained to, enabling it to move to any position on a flat substrate.

The difference to the demonstration in Section 7.6.1 above is that the four-legged

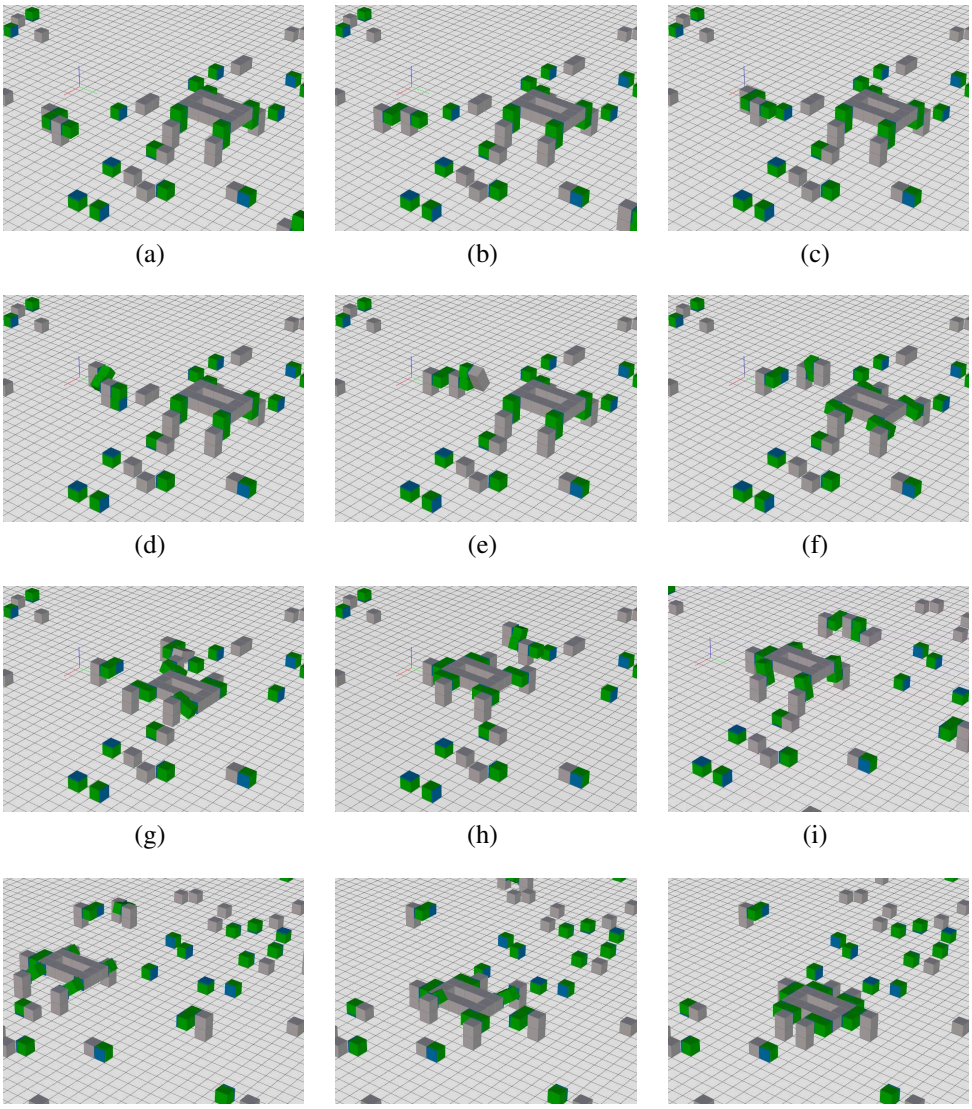


Figure 7.23: Sequence of screenshots of the "predatory behavior" demonstration in simulation. A four-legged robot acquires two additional legs from smaller robots independently capable of locomotion.

robot does not grow the legs from individual modules harvested from the environment, but from smaller robots (Figure 7.23(a) to 7.23(h)).

7.7 Hardware Design Extensions

While the set of three module types, actuation, structural, and energy, is sufficient to implement complex machines and behaviors that form an ecology of robots, the design of Soldercubes is extensible. This allows for more versatile applications of the concept of the robot ecology, and modular robotics applications in general. The following sections describe three module types that have been constructed to illustrate further extensions of the Soldercube design into a multi-purpose modular robot system.

7.7.1 Towards Untethered Operation: Wifi Module

In all experiments with Soldercubes described so far, all modules are physically connected to a common communication bus through other modules and the substrate tiles. Many applications do, however, require the operation of various system parts in locations that might not be on one substrate. Even when operating in close proximity, the creation of a continuous substrate might be prohibitively costly or invasive. For example, the physical implementation of the simulated predatory behavior in Section 7.6.2 would have required 40 modules (with a total of 240 connectors), and a substrate sized 20 by 20 tiles, i.e. nearly twice as many connectors in the substrate as in the robot!

The Wifi connectivity Soldercube module extends the structural module to house an electricimp™ wireless network node. The electricimp imp001 is a fully integrated IEEE802.11 b/g/n WiFi including antenna and microcontroller in the form factor of a

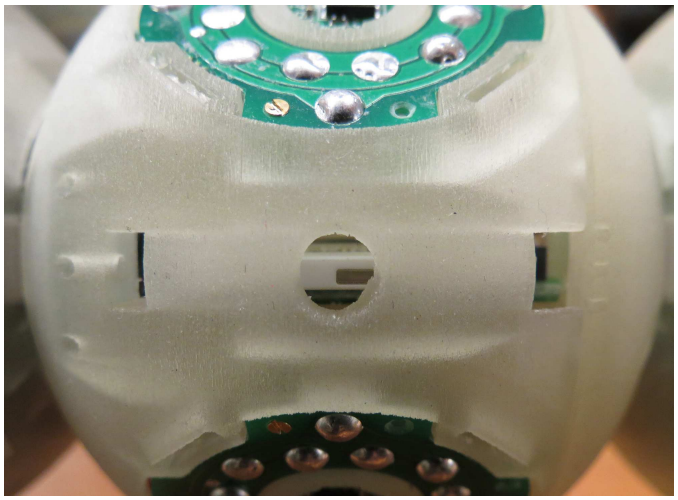


Figure 7.24: Modified module shell for the Wifi module. A round center cutout gives access to the electricimp photodiode used for transmitting network access credentials optically. Two square cutouts are to accommodate the physical dimensions of the electricimp development board.

Secure Digital (SD) card [183]. Each electricimp connects directly and exclusively to an internet server operated by Electric Imp, Inc. through which the module is globally uniquely addressable. The initial connection setup with the local wireless network is through a process named "blinkup" whereby the access credentials are transmitted optically through a flashing smartphone screen to a photodiode on the electricimp.

At the time of designing the Wifi module, only the electricimp imp001 module was available, which requires an SD card socket and cryptochip with globally unique ID as external components to be supplied. To reduce development time, the commercially available "April" developer board for electricimp which contains these components was used to construct the Wifi module. Because the physical dimensions of the April PCB exceed the maximal linear dimension available inside the cube shell by 2 mm, the shell of the Wifi module was modified with cutouts to accommodate the PCB as shown in 7.24. To provide access to the photodiode which is used for optically passing network access credentials to the electricimp during initial setup a further round hole is cut from

the cube shell.

The electricimp's operating voltage is 3.3 V while the Soldercube logic level voltage is 5 V. The electricimp can therefore not be directly connected to the Soldercubes communications bus. Instead, the Wifi module's microcontroller acts as a router, sending all data received from the communications bus to the electricimp, and vice versa. The connection between microcontroller and electricimp is over UART using the second UART port of the main microcontroller on the Soldercube control PCB. In actuation modules this is used for communication between the main controller PCB and the motor control PCB, which means that even disregarding space constraints an actuation module could not easily be extended with wireless networking capabilities. The baud rate is chosen as a multiple of the data rate on the communication bus.

The electricimp is programmed to buffer all data bytes received from the Wifi module controller and continuously forward the buffer contents as serialized payload in HTTP POST requests to its associated internet server. Requests are send synchronously, i.e. only one outgoing HTTP request is open at any time, to preserve the time order of data. Data is not parsed on the electricimp making it a protocol-agnostic component of the communication system. Incoming HTTP messages are unserialized and placed in the electricimps UART output buffer for transfer to the Wifi module controller.

The server associated with the electricimp™, referred to as "agent" in the electricimp™ specifications, is programmed to forward any data it receives from the agent to other registered electricimp agents (from other Soldercubes Wifi modules). This means that to connect any two Soldercubes Wifi modules one must register their respective electricimp agents with each other, currently a manual step.

The immediate forwarding of data from the electricimp server agent to the elec-

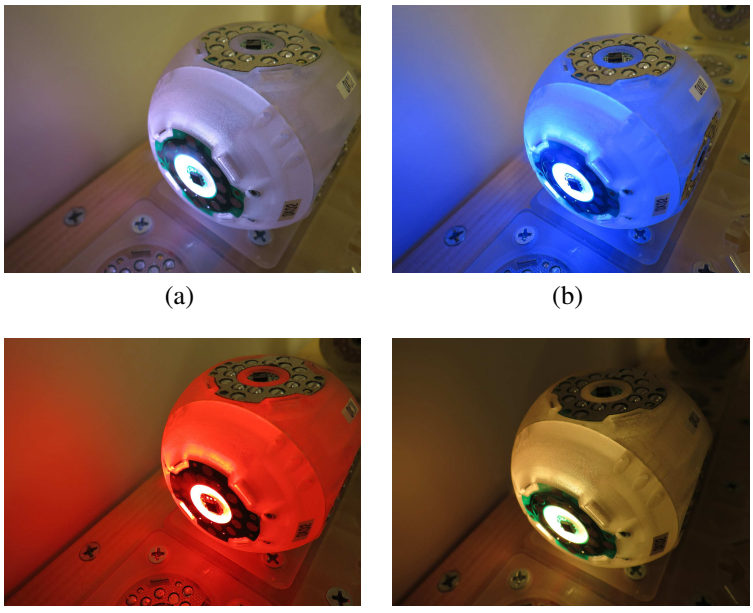
tricimp hardware in a Wifi module is only possible due to the implementation of the connection between the two entities. Normally, devices connected to a wireless network access point cannot be directly reached from an internet server. This means that while electricimp server agents can send messages to their associated electricimp hardware, it is not easily possible to push messages to the control software running on a PC. To overcome this problem, any incoming data is buffered temporarily by the electricimp agent. Control software running outside the module assembly has poll each associated electricimp agent for incoming data.

7.7.2 Towards Signal Output: Light Module

The light module is a structural module extended to hold a high brightness RGB color light emitting diode (LED). The module, shown in Figure 7.25, re-purposes the microcontroller pins reserved for communication with the servo motor controller PCB in the actuated module and uses them to generate three pulse-width modulated control outputs for each color channel of the LED. Power is drawn directly from the 12 V power supply line available in each Soldercubes module.

The motivation to construct the light module was the lack of a high bandwidth human readable information channel during testing and development. This was particularly useful when sending of debug messages through the communications bus would have interfered with normal communication or when the communication system was unavailable. Instead of the binary visual output available from the two miniature LED in each module, the light module can produce light in any sequence of visible colors.

Beyond debugging, further applications could be as a communications channel with yet to be developed sensor modules, as beacons. Examples for such beacon applications



(a) (b)

(c) (d)

Figure 7.25: Photographs of the Soldercube light module

of light emitting robot modules are the use for self-assembly of swarm robots by Gross et al [49] and for robot localization and alignment as demonstrated with CKBot by Yim et al [50].

7.7.3 Towards the Analog World: Wheel Module

Because the actuation module is constructed to support infinite rotations, and the communication protocol language described in Section 7.3.2 contains instructions for continuous rotation, the actuation module can be modified to support various other functions that require continuous rotation. Implementing a wheel module converts the lattice type modular robot system that operates in a discretized world, as described in this chapter so far, into a mobile modular robot that operates in a continuous fashion.

To demonstrate that the resulting mobile modular robot a car structure was assem-

bled with four wheel modules and a "body" with one energy module, one Wifi module, and two structural modules. The car moves on a surface to a single light module placed in the environment. By heating its appropriate connector while adjacent to the LED module, the car robot acquires the LED module from the environment.

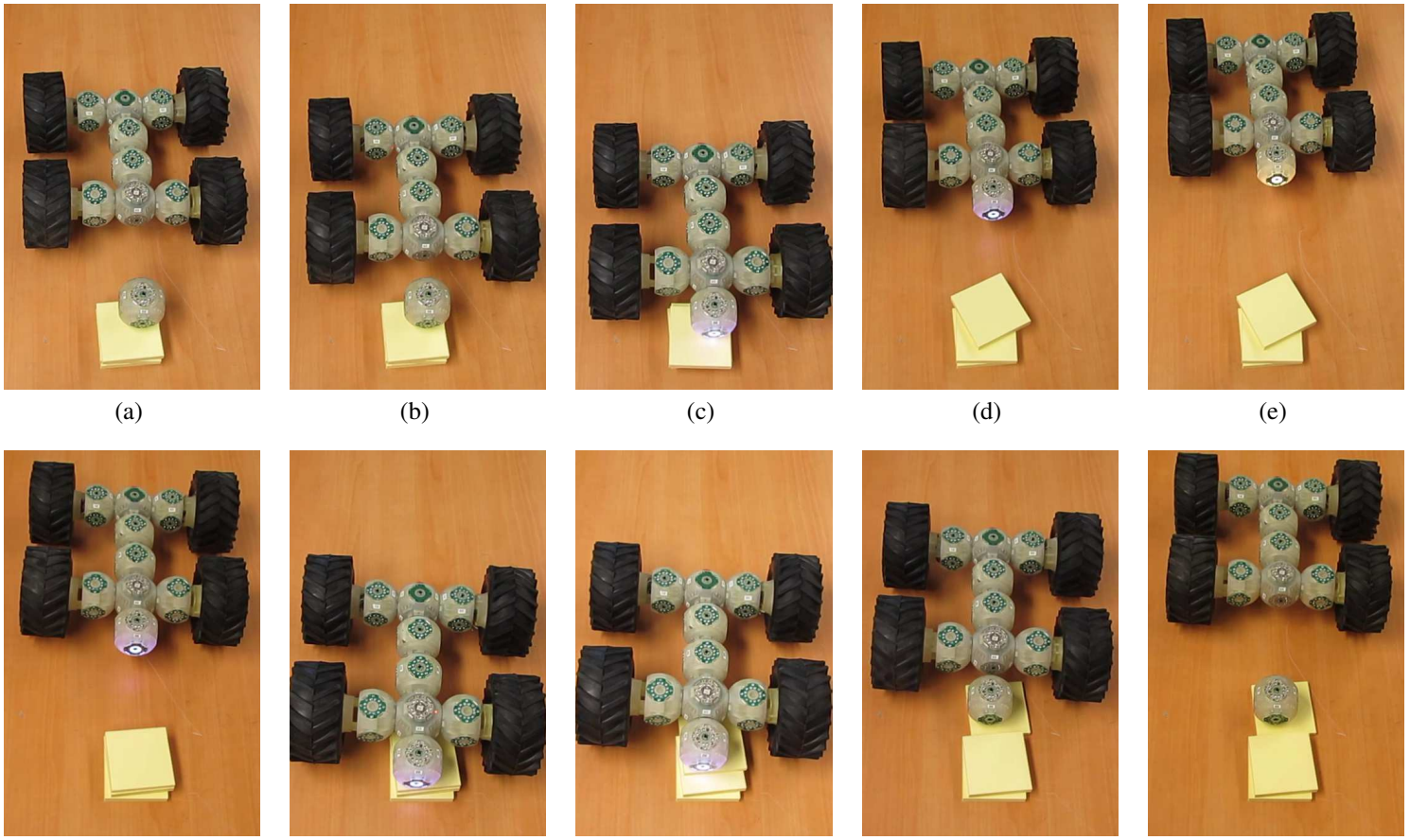
As soon as the electrical connection to the newly acquired module is established the LED module initializes into a default state emitting blue light. Control of the newly acquired module is possible from any other cube or by facilitating the Wifi module's function. In this experiment, the GUI is used to send alternating commands to the light module at regular intervals resulting in flashing. Figure 7.26 shows this scenario in a sequence of photographs.

This experiment demonstrates that all functions of the Soldercube system including self-reconfiguration and acquisition of modules from the environment are available when in a mobile configuration with wheel modules. The concept of a robot ecology with multiple interacting robots who exchange modules between each others applies to mobile modular robots. This experiment shows the basic case of a robot acquiring a new function, emitting light signals, by autonomously picking up a module from its environment.

7.8 Conclusions and Directions for Future Work

7.8.1 Achievements

The Soldercube system is a successful attempt at creating a self-reconfiguring modular robot system with high manufacturability. By incorporating the self-soldering connec-



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 7.26: Sequence of photographs of the “car headlight” experiment. A wheeled robot with one energy module and one Wifi module moves untethered. It acquires a module from the environment to add functionality, and later returns it back into the environment.

tor, and making use of manufacturing techniques and electronics products that have only in recent years become available for low volume applications, the complexity, size and weight of the module are smaller than for any comparable modular robot system. Soldercubes is also among the few existing self-reconfiguring modular robot systems for which more than ten of modules have been constructed and used in an assembly, proving the claim for scalability by example.

Several demonstrations were presented, partially in physical experiment and partially in simulation, that demonstrate complex ecosystem-like interactions between modular robots. These may act as an indication that he strategy of producing large numbers of simple modules is a viable strategy towards building reconfigurable machines that are flexible to adapt to various tasks and environments.

By using the ecosystem analogy, Soldercubes have demonstrated the exchange of robot parts between robots, and between robots and the environment. While it is common for groups of robots to exchange information, and the exchange of energy has been demonstrated, the exchange of matter is so far a unexplored aspect of multi-robot systems.

7.8.2 Lessons Learned: Suggested Hardware Design Improvements

While the Soldercubes system has been used successfully to demonstrate several functions, all demonstrations required manual intervention and the reliability of Soldercubes robots is low. When each individual module works reliably for 95 % of operations, say, by trivial statistics a robot consisting of 10 actuated Soldercubes would only function correctly for 60 % of operations, and a robot with 40 modules only in 12 % of operations. The fundamental solution to this problem is to move towards using design and manufac-

turing methods suitable for high volume production instead of prototyping methodologies. This was attempted in part for Soldercubes: Many electronics components rely on small scale surface mount designs like they are found in consumer electronics products, and those components were in fact rarely the cause of failure in experiments. Other aspects of the Soldercubes design do, however rely on manual labor and are error prone, ultimately being the cause of insufficient reliability for large scale experiments.

As a means of providing guidance for future attempts at constructing large scale self-reconfiguring modular robot systems, the following is a short review of design improvements that are suggested before more complex demonstrations are attempted. These are lessons learned pertained to implementation details and none does invalidate the general claim that the Soldercube module is a step towards scalabele to large systems of tens or even hundreds of modules.

The primary source of error in the electrical module design were the low insertion force connectors for flat flexible cables. These components were a frequent source of short circuits or connections between the signal line and ground which is likely due to the shear force exerted by the connector onto the cable resulting in loose copper traces on the cable. While it might be possible to overcome this failure mode through selecting other components, a redesign towards manufacturing the main controller PCB and connector PCBs as one flat-flexible PCB is suggested. This would have the additional benefit of freeing up space on the main controller PCB which can be used for additional functionality, or further volume reduction, or for using larger package size components for easier assembly.

For the actuator module, the use of an integrated servo motor package proved to be more burden than help. While the idea to rely on a proven closed system for motor control seems attractive, it can be said with the benefit of hindsight that full control

over a custom system is preferable. The primary problem with the Dynamixel servo motor product used is the lack of low level control, which combined with unpredictable potentiometer behavior in the dead band led to numerous reliability problems. For future implementations the design of a motor drive and control system from ground up would be suggested.

The mechanical properties of the module shell were limited by the fabrication employed available for prototyping. Especially when assembled into cantilevered constructions of two or more Soldercubes, it becomes apparent that both the stiffness of the module shell and the strength of screw connections inside the module shell is too low. For future implementations it is suggested to machine the module shell out of a material stronger than the 3D-printed Object Fullcure 720 material used in the present implementation.

CHAPTER 8

CONTRIBUTIONS

8.1 Major Contributions

Chapter 3 A novel connection method for self-reconfiguring modular robots based on forming soldered joints between robot modules was developed. The connector requires no external manipulation when forming a connection, is mechanically strong, reversible, electrically conductive, and only requires power when connecting and disconnecting. It is fully contained on a single printed circuit board, has no moving parts, weighs only 2 g and has a total thickness of under 3 mm. This makes it the lightest, least complex, and likely cheapest connection method for modular robots to date. Two reference implementations, one in a 14 mm square shape and one in an approximate annulus shape with a surface area of approximately 1 in were implemented and later used in Chapters 6 and 7. The mechanical strength and repeatability of the connection method were investigated in experiments. In addition, a thorough explanation of component selection and the assembly process has been provided to serve as a guide for future application specific implementations.

Chapter 4 The concept of the thermorheological valve which has been previously used at the microscale has been applied at the mesoscale. After an investigation of available liquids with suitable thermorheological properties, a mixture was selected that provides properties superior to those previously used for self-assembly experiments. A thermorheological valve was developed that is able to stop flow at rates up to $1.0\,\mathrm{g\,s^{-1}}$ at pressures up to 0.9 psi, and reduce the flow rate of flow at applied pressures up to 1.3 psi. The valve's utility was demonstrated in a simple flow routing experiment where it was

used to route flow between two outlets.

Chapter 5 A minimally simplistic communication architecture for small low complexity modular robot systems was presented and experimentally confirmed as functional.

Chapter 6 Two module designs for a stochastic fluidic assembly system were designed and constructed. The first system contains six soldering connectors as described in Chapter 3 and implements internal flow routing using six thermorheological valves as described in Chapter 4. The second design replaces the thermorheological valves with six custom built miniature solenoid valves. Appropriate assembly media were selected for each module, and an assembly chamber and assembly substrate for the latter were implemented and tested with a passive module.

Chapter 7 The Soldercubes self-reconfiguring modular robot system was developed. Soldercubes modules operate in a lattice configuration, have one actuated rotational degree of freedom and six soldering connectors as described in Chapter 3. An actuated Soldercubes module weighs 120 g and has a side length of 55 mm making it smaller and lighter than any modular robot system with comparable feature set. In addition to the actuated module, a structural module and a rechargeable energy module that can provide power to Soldercube robots were developed. Due to the use of the soldering connector, Soldercubes modules are less complex than other modular robot modules and systems are more readily scaled to larger module counts. 40 Soldercubes modules were constructed and experiments with 27 concurrently operating ones were shown, making it one of the few known self-reconfiguring robot systems whose scalability was proven by implementing more than 10 modules. By using the Soldercubes system, the

concept of a robot ecology was introduced where robots not only exchange information and energy, but also matter, or robot parts, in an experiment. Further, more complex, robot ecology behaviors were simulated and could be implemented once Soldercubes modules are manufactured in a more reliable fashion.

8.2 Contributions of Others

Chapter 3

- Abraham Cantwell collaborated on the design and experimental validation of the first iteration of the self-soldering connection method.
- I collaborated with visiting researcher Arne Rost on conducting repeatability and stress tests of self-soldering connection method.

Chapter 4

- The design of the thermorheological valve is based on prior work and incorporates suggestions by graduate students Michael Kolantarov, Mekala Krishnan, and Prof. David Erickson.
- Prof. Heinrich Jaeger and his graduate student Eric Brown at the University of Chicago performed the quantitative experiments described in Section 4.3.2 and generated Figure 4.2, and contributed to the design of the thermorheological valve.

Chapter 5

- B.Eng. student Jeremy Blum assisted with design and testing of printed circuit boards and embedded code.

Chapter 6

- B.Eng. students Jeremy Blum and Abraham Cantwell and research assistant Stephane Constantin assisted with the electronics design of printed circuit boards and embedded code development.

Appendix A

- Sebastian Risi and Nicholas Cheney provided helpful suggestions on the usage of NEAT and HyperNEAT.
- Daniel Ly provided helpful suggestions on evolutionary search methods in general.

APPENDIX A

EXPERIMENTS IN DESIGN AUTOMATION

The demonstrations of robotic ecosystem interactions in Chapter 7 are preplanned manually in order to demonstrate the flow of matter through an ecosystem of modular robots. The motivation for implementing ecosystem interactions in robots is to allow a group of robotic machines to adapt flexibly to changing requirements. To realize the full potential of this flexibilty, interactions should be planned autonomously based on the current state of the environment, and not through time consuming manual definition of each modules movements. This chapter is intended to be a brief exploration of evolutionary search techniques to the problem of finding controllers for self-reconfiguring modular robots.

A.1 Control of Modular Machines

In realistic application scenarios of self-reconfiguring robots, the location of any given module is not known a priori. Instead it is determined by the time history of the module's interaction within the environment as well as the interactions of all other modules in the environment. In our system, a module could potentially have been part of an assembly with locomotion capabilities before having been individually disposed into the environment, from where it was assembled into yet another machine with locomotion capabilities.

While the position of modules in a modular machine is usually more predictable in laboratory experiments, we strive to find controllers that are suitable for realistic use cases including those of machines which acquire modules from and reject modules into the environment. This leads to the requirement that every module must operate its own

instance of the control algorithm. A central machine-wide controller for module assemblies would be infeasible for many reason, most fundamentally the lack of existence of any part of the machine that is guaranteed to remain part of the same machine in future. Behaviors of assembled machines then emerge from the interactions of individual module controllers.

Haasdijk et al. [184] similarly argue for distributed control between modules in modular robots, and conclude that specialized behaviors of modules depending on their position within a machine are desirable. Hence, they suggest using a generative encoding universal across all modules, which generates a family of distinct controllers. The modules' positions within the robot are used as inputs for the generator which is implemented as a CPPN. More precisely, the position input for the generative encoding in [184] is an arbitrarily scaled two dimensional geometric position of the module in an arbitrarily chosen robot configuration. While this method is conceptually attractive, the implementation presented in [184] relies on fixed robot morphology, a well-defined initial (or otherwise "natural") configuration, and knowledge of global position of modules within a robot. In particular, the methodology does not support machines of unknown size or machines which can exchange modules with the environment during operation; neither is there a clear path to extend the method to these use cases. Using a generative encoding with relative positions of modules as inputs is therefore not feasible for a self-reconfiguring system.

The approach here is to not supply any global location information to the module. Instead we allow modules to pass a limited amount of information between each other (one floating point activation signal). In additions, modules can sense presence or absence of connected neighbors. Both assumptions are realistic and scalable with our physical system and other self-reconfiguring modular robot hardware built to date

(for example [20, 81, 88]). Given these inputs, both local specialization and cyclic behaviors are possible. This would be in a fashion similar to hormone based control for self-reconfiguring modular robots [70]. However, instead of using control tables to determine actuation from signal inputs , we investigate the use of artificial neural networks as controllers.

A.2 Artificial Neural Networks as Module Controllers

Each module's controller is implemented as a three layer feed-forward artificial neural network (ANN) with 12 inputs, 6 hidden layer nodes, and 3 outputs. The activation functions of all neurons are the hyperbolic tangent (sigmoid) function resulting in the following simple transfer function for each neuron:

$$\text{out} = tanh(\sum_{\text{inputs}} \text{weight}_i \cdot \text{input}_i) \tag{A.1}$$

A graphical representation of the network layout is shown in Figure A.1.

The six binary inputs represent the presence of neighbors in the three-dimensional von Neumann neighborhood, i.e. the presence of connected modules on all six connection surfaces of the module. An additional six input signals represent floating point signals in the interval [-1.0, 1.0] transmitted from the same neighboring modules, if present. A detailed description of all input values is given in Table A.1, the outputs are given in Table A.2. A bias node with fixed value of 0.5 was further added to the input layer.

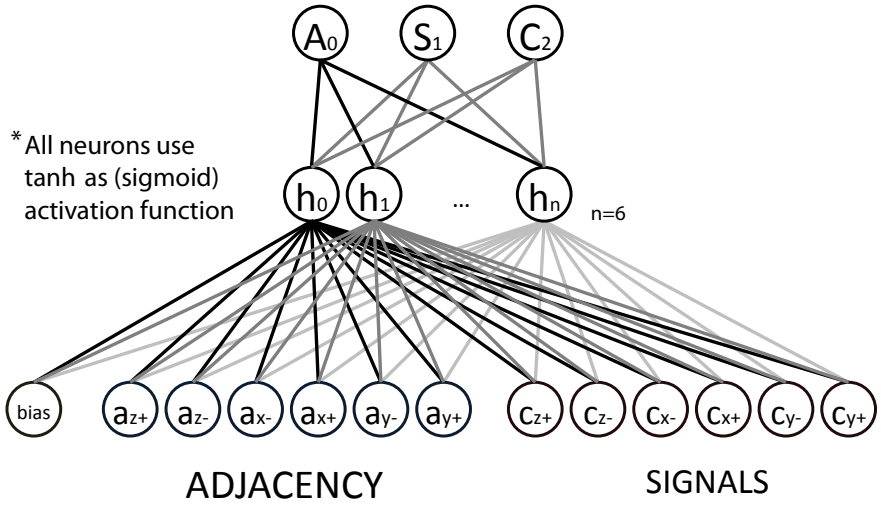


Figure A.1: Neural network controller layout. The controller operating on each module is a neural network with 12 inputs (adjacency and signals from neighboring modules), 6 hidden layer neurons and 3 outputs.

Table A.1: Artificial Neural Network Input Descriptions

Label	Values	Description
$a_{n+/-}$	{-1, +1}	Adjacency: Positive valued when a neighboring connected module is present in the positive/negative n-direction (where n is [x,y,z]), otherwise negative valued.
$c_{n+/-}$	[-1.0, 1.0]	Signals: If a neighboring module is present in the positive/negative n-direction, its S output is fed into the network as floating point input.

Table A.2: Artificial Neural Network Output Descriptions

Label	Values	Description
A	[-1.0, 1.0]	Actuation: Output value is discretized into 8 bins, each defining on actuation pattern
S	[-1.0, 1.0]	Signal: A floating point value which can be read from connected neighboring modules.
C	[-1.0, 1.0]	Connectivity: A value defining the module's affinity towards the substrate. Of the modules adjacent to the substrate, that with the highest connectivity value forms a connection to ground, while all others remain free to move.

A.3 Evolutionary Search

While the topology of the controller ANN remains fixed, the weights of neural connections is optimized using a genetic algorithm. The following describes the fitness function used for each search function, followed by a description of the two evolutionary search methods investigated.

A.3.1 Fitness Functions

The overarching goal of this investigation is to find controllers for driving the entire behavior of modular machines including locomotion and self-reconfiguration, or more generally: any interactions with the environment. Here we restrict ourselves to finding controllers for locomotion.

The phenotype used for fitness evaluation is a kinematic simulation of the modular machine in a simulator. The controllers described in Section A.2 are executed for every module in the simulated world and respective actuations are applied and geometrically simulated. For this, the simulator described in Section 7.4 is used in headless mode with a complete simulation environment being created, executed, and torn down, for each fitness evaluation. The purpose of the simulator is not to simulate continuous physics but only to compute the geometrical arrangement of modules in space at any time. It is further assumed that actuation is in 90° intervals only such that all modules remain aligned with the world grid after any set of actuation commands is executed. This is a true representation of our physical prototype system.

Maximal Locomotion

The naïve implementation of a fitness function to reward locomotion is to measure the distance travelled in a given amount of simulated time, for example 20 s:

$$d = x_{c1}(t = 20) - x_{c1}(t = 0) \tag{A.2}$$

As the simulation used does not simulate physics, such a fitness function is prone to discover loopholes in the simulation and yield physically impossible gaits. The most obvious such loophole is that our simulator allows the robot to move through the ground plane. This could be addressed through a more complex simulator which simply rejects such movements, or by a fitness function that penalizes intersections between modules and the ground plane.

The simplest penalty against physically impossible phenotypes is to assign a fitness of zero. This would, however, dramatically reduce the evolvability of the problem as no gradient towards physically possible solutions exists. The solution chosen here instead assigns an efficiency value for every phenotype based on the path distance travelled by each module, and modifies this measure by a factor of plausibility based on which part of the simulated world the module moves in.

Efficient Locomotion

Instead of simply rewarding the distance travelled, the efficiency fitness metric rewards distance travelled per effort. Effort is measured as the average squared path distance travelled by every cube over the simulated time frame:

$$\omega = \frac{1}{N} \sum_{t=1}^{n^t} \sum_{\text{cubes}}^{N} \eta (x_t - x_{t-1})^2 \tag{A.3}$$

where the number of cubes is N and the simulation completes n_t simulation steps. This can be thought of as a simple estimate for the work required to move the machine.

The factor η is introduced to penalize specific undesirable behaviors: Intersections of robot and ground plane, and tall robot configurations. The fitness is then computed by dividing distance travelled d and effort w.

A.3.2 Genetic Algorithm

With the controller represented as ANN, several well known solution encodings are available, two of which are used in this work. Firstly, the solutions are represented as an ordered vector of ANN weights which are operated on by a genetic algorithm. Secondly, the solutions are encoded using the HyperNEAT algorithm's generative encoding.

Age-Fitness Pareto Selection

Age-fitness selection as described by Schmidt et al. in [185] was used to search the space of 96-element floating point vectors representing the ANN connections weights. Crossover is by simple 2-point crossover, mutations are point mutations of normally distributed numeric magnitude. At every generation all members of the current generation are paired up to generate offspring, followed by a tournament based selection. In addition, one randomly generated new individual is introduced each generation. Each solution has an associated age which is incremented at each generation step. Offspring inherit the age of their parents. During the selection step, tournaments of two individu-

Table A.3: Substrate locations of ANN nodes

Node label	**x**	**y**	**z**	**n**
a_{x-}	-1.0	0	0	-1.0
a_{x+}	+1.0	0	0	-1.0
a_{y-}	0	-1.0	0	-1.0
a_{y+}	0	+1.0	0	-1.0
a_{z-}	0	0	-1.0	-1.0
a_{z+}	0	0	+1.0	-1.0
c_{x-}	-1.0	0	0	+1.0
c_{x+}	+1.0	0	0	+1.0
c_{y-}	0	-1.0	0	+1.0
c_{y+}	0	+1.0	0	+1.0
c_{z-}	0	0	-1.0	+1.0
c_{z+}	0	0	+1.0	+1.0
bias	0	0	0	+1.0
h_0	-0.5	0	0	0
h_1	+0.5	0	0	0
h_2	0	-0.5	0	0
h_3	0	+0.5	0	0
h_4	0	0	-0.5	0
h_5	0	0	+0.5	0
o_0	0	0	0	0
o_1	0	0	0	+1.0
o_2	0	0	0	+0.5

Table A.4: Parametric values for the NEAT Algorithm

Parameter	**Value**
AddConnectionMutationProbability	0.075
DeleteConnectionMutationProbability	0.075
AddNodeMutationProbability	0.05
ConnectionWeightMutationProbability	0.8
ConnectionWeightRange	1.0
InitialInterconnectionsProportion	0.6

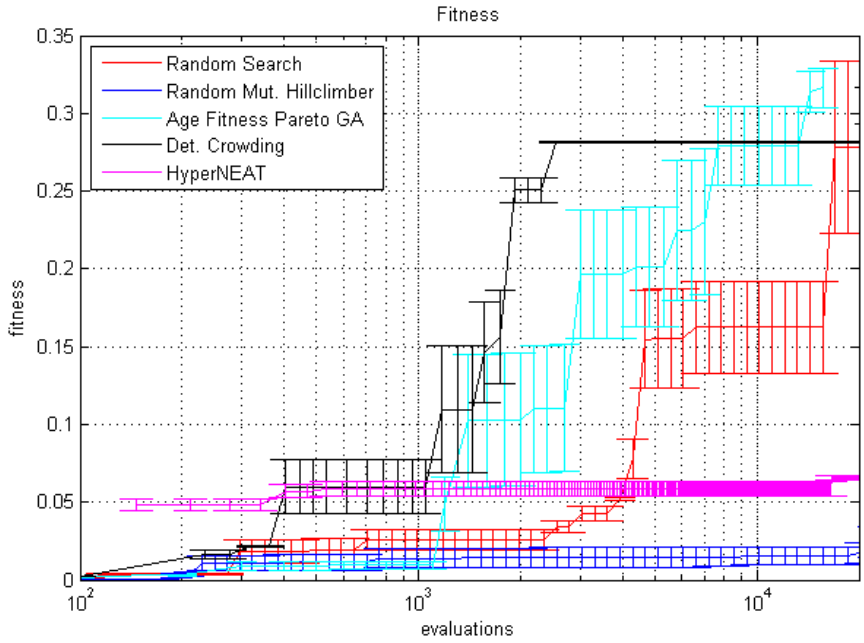


Figure A.2: Fitness per number of evaluations for the efficiency based fitness function. Each line shows the average of three experiments, error bars show standard error.

als are formed until the population size is reduced to the target population size of 100. Elimination only occurs when one individual in a tournament dominates the other by both fitness and age.

HyperNEAT

HyperNEAT is the method of obtaining the weights of an ANN's connections from an evolved Compositional Pattern Producing Network (CPPN) [186]. CPPNs are a superset of ANNs: In addition to sigmoid CPPNs also emloy other types of functions as activation functions. The CPPN is evolved using the Neuroevolution of Augmented Topologies (NEAT) algorithm which finds both topology and weights of the network [187].

When an evolved CPPN is queried with coordinates of two ANN nodes in an arbi-

trary user-defined space it returns the weight of the connection between them. To gain the benefits of this generative encoding, which include regularity and symmetry [187], a suitable spatial arrangement of ANN nodes in space is required. To this end, we placed the input nodes on a unit sphere around origin at 90° angles to each other to resemble their spatial arrangement in the physical module. A fourth dimension was introduced to differentiate inputs of different types (signal vs adjacency). Similarly, hidden layer neurons are placed around the coordinate origin on a half-unit sphere. Outputs are all placed at the origin, as is the input bias neuron, but offset from each other in the fourth dimension. The exact spatial positions of each ANN node are given in Table A.3. Other relevant NEAT parameters are listed in Table A.4.

Baseline Comparison

As baseline comparisons for this search we used Random Search (RS), Random Mutation Hill Climber (RMHC), and Deterministic Crowding (DC) which is commonly used as a benchmark for pareto-selection based genetic algorithms. RS generates a random candidate solution is generated at each evaluation and stored if it beats the previous best candidate solution. RMHC applies the same mutation function as the Age-Fitness Pareto GA to the existing best candidate solution and replaces this if the changed solution is better or equally good. DC is with a population size of 100 and using the same mutation and selection functions as Age-Fitness Pareto Evolution.

A.4 Results

Figures A.3 to A.5 show a selection of best individuals obtained from Age-Fitness Pareto selection algorithm searches. Figure A.3 shows a behavior that exhibits a walking gait

travelling approximately one body length during 10 seconds. Only one side of the robot contributes to locomotion while the legs at the other side do not support the robot. This behavior and similar ones are obtained when explicitly penalizing motions resulting in tall robot configurations.

Figure A.4 shows a behavior obtained from a search using the distance-only based fitness function: Here the gait results in the machine flipping over repeatedly. While this is indeed a very fast way of moving which outperforms walkers designed by hand, it is not a desirable gait because it would induce large stresses in the structure, a drawback not captured by the distance-only based fitness function.

A result conceptually intermediate between the previously explained two is shown in Figure A.5. Here an efficiency-based fitness function was applied but no penalization for either intersecting the ground plane or tall robot configurations was added. The resulting gait flips over but the machine remains as compact during the rotation in order to reduce the distance travelled by modules (thereby increasing the efficiency of the gait).

Figure A.2 shows the search progress of the various search techniques investigated. Notably, RMHC performs poorly suggesting a fitness landscape with many local optima. It is outperformed by DC and Age-Fitness pareto selection based genetic algorithm, the latter of which continues to find improved solutions after DC plateau-ed showing the expected benefits of introducing diversity into the population at every generation step. Surprisingly though, random search performs comparably well (yet less reliably so) as genetic algorithm search - a result that warrants for further investigation.

HyperNEAT does not perform favorably as a search method. However, the NEAT generative encoding appears to produce fitter individuals than randomly chosen weight vectors, as is exhibited by the initial random population performing consistently orders

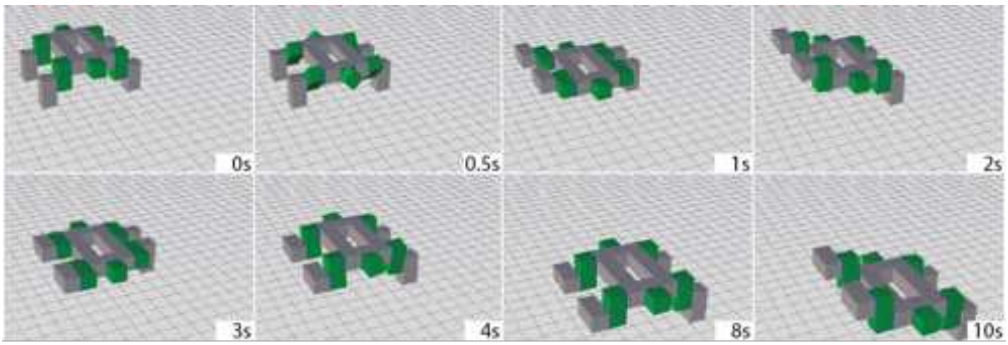


Figure A.3: Candidate solution found from Age-Fitness Pareto selection algorithm. The behavior exhibited shows a walking gait on one side of the robot while the legs at the other side do not support the robot.

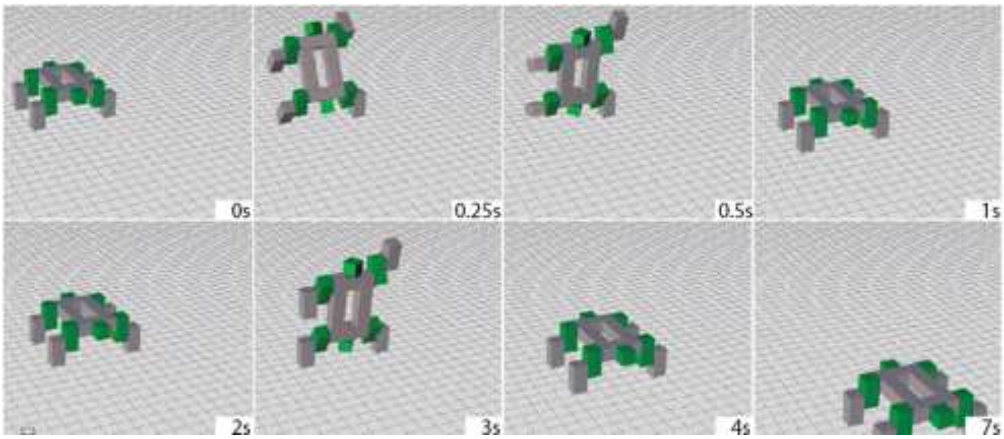


Figure A.4: Candidate solution found from optimization with distance-only fitness function. When efficiency is not included in the fitness computation, flipping over in a "cart wheel" fashion is the most successful gait.

of magnitude better than random individuals using the direct encoding. Not shown are visualizations of results from the HyperNEAT based evolutionary searches. These do in fact exhibit an increased regularity and symmetry in the gaits, with both robots sides frequently moving symmetrically and in synchrony.

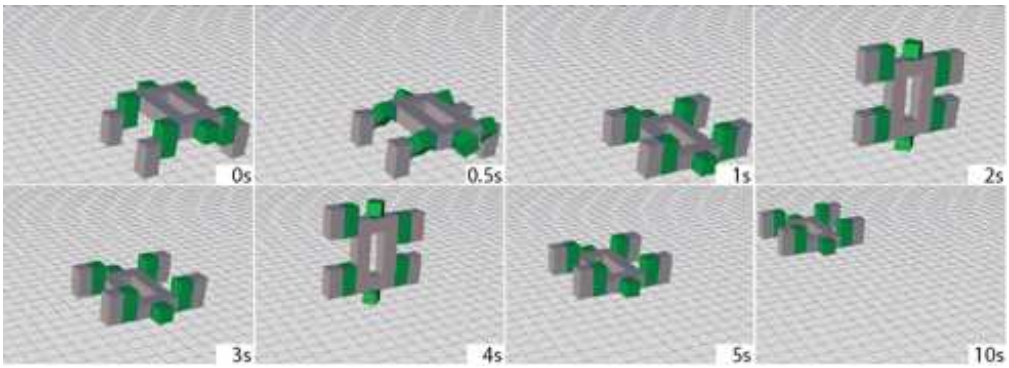


Figure A.5: Candidate solution obtained by random mutation hill climber with efficiency included in the fitness function. A similar solution as in (b) was initially discovered and was optimized locally to reduce path length travelled by cubes. The effect of this can be seen from the frames at 2s and 4s where the robot is not fully extended but compressed to be as small as possible.

A.5 Discussion

The work presented in this chapter has shown that distributed locomotion controllers for modular machines can be found using simple optimization techniques such as RMHC and more efficiently by genetic algorithms. In addition, using HyperNEAT did result in more regular and symmetric gaits than other search techniques, despite not using the location of a module in the machine as an input. This could be attributed to the fact that the layout of ANN nodes on the CPPN substrate resembles the spatial distribution of inputs. This hypothesis was, however, not validated as part of this work.

The best controllers discovered by all search algorithms including the RMHC regularly outperformed hand-designed controllers in terms of distance covered. However, the behaviors are rarely desirable because they would either induce large stresses in the connections between modules, or require large torques, or both. The primary reason why search techniques did seldom find robot behaviors similar to humand-designed ones is that the formulation of a fitness metric that accounts for the physical limitations

of the robot while still rewarding locomotion is surprisingly difficult.

Given the promising result that the design of controllers for locomotion of modular robots proved possible with genetic algorithms, it seems suitable to extend the technique to more complex tasks including interaction with the environment through connection and disconnection of modules. NEAT encoding appears suitable because it produces more regular behaviors but it needs to be validated whether this persists for other tasks or is a problem-specific result.

APPENDIX B

BILL OF MATERIALS

This Appendix lists the bill of materials for all robot modules describe in Chapter 7 in the *Robot Ecologies* application. Tables B.1 to B.3 are top level bills of material, referencing second level listings in Tables B.4 to B.7 for all in house designed sub-assemblies.

Pricing information is as of December 2012. Where applicable, the unit price is given for individual purchase and batch purchase of 50 items, the latter as an indicator for volume level pricing when a collection of robot modules is to be built.

Table B.1: Bill of Materials, Actuation Module

Name	**Qty/Module**	**Source/Reference**
Outer Shell Large Part	1	3D-printed (Polyjet)
Outer Shell Small Part	1	3D-printed (Polyjet)
D1/16 in L1/4 in Magnet	24	KJ Magnetics D14-N52
Controller PCB	1	Table B.4
Connector PCB	6	Table B.5
Sensor PCB	4	Table B.6
Electric Motor	1	from Dynamixel AX-12A
Motor Controller PCB	1	from Dynamixel AX-12A
Potentiometer	1	from Dynamixel AX-12A
Potentiometer Idler Gear (29 teeth)	1	3D-printed (Polyjet)
Spur Gear Pair (16, 52 teeth)	1	from Dynamixel AX-12A
Spur Gear Pair (9, 31 teeth)	1	from Dynamixel AX-12A
Spur Gear Pair (11, 22 teeth)	1	from Dynamixel AX-12A
Spur Gear (28 teeth)	1	from Dynamixel AX-12A
Nylon Bushing	2	from Dynamixel AX-12A
D2 mm L22 mm Dowel Pin	1	from Dynamixel AX-12A
D1.6 mm L15 mm Dowel Pin	1	from Dynamixel AX-12A
M1.0 L8 Dowel Pin	1	McMaster 93600A033
Internal Gear (69 teeth)	1	3D-printed (Polyjet)
Thin Section Bearing	1	Impact Bearings IM-KAA15XLO
12 Strand Slip Ring	1	Moog SRA-73540-12
Slip Ring Cap	1	3D-printed (Polyjet)
M2.0 Torx 5 Thread-cutting Screws	20	McMaster 99397A314
Miniature Screws	30	JIMorris Co F000120B094

Table B.2: Bill of Materials, Structural Module

Name	Qty/Module	Source/Reference	Man Part No
Outer Shell Large Part	1	3D-printed (Polyjet)	
Outer Shell Small Part	1	3D-printed (Polyjet)	
D1/16 in L1/4 in Magnet	24	KJ Magnetics	D14-N52
Controller PCB	1	Table B.4	
Connector PCB	6	Table B.5	
Sensor PCB	6	Table B.6	

Table B.3: Bill of Materials, Energy Module

Name	Qty/Module	Source/Reference	
Outer Shell Large Part	1	3D-printed (Polyjet)	
Outer Shell Small Part	1	3D-printed (Polyjet)	
D1/16 in L1/4 in Magnet	24	KJ Magnetics	D14-N52
Controller PCB	1	Table B.4	
Connector PCB	6	Table B.5	
Sensor PCB	6	Table B.6	
Battery Charging PCB	1	Table B.7	
Battery Holder PCB	1	custom fabrication	
RCR123A 3.2 V Rechargeable Battery	3	various vendors	

Table B.4: Bill of Materials, Main Controller Printed Circuit Board.

Description	Qty/ Module	Value	Parts	Manufacturer	Man Part No	Vendor	Price	Price*
Resistor	1	1 MΩ	R1	Rohm	MCR01MRTJ105	Digikey		0.01260
Resistor	4	1 kΩ	R8, R9	Rohm	MCR01MRTJ102	Digikey		0.01260
Resistor	2	4.02 kΩ	R3, R4	Panasonic	ERJ-2RKF4021X	Digikey	0.10000	0.02420
Resistor	3	10 kΩ	R2, R10, R11	Rohm	MCR01MRTJ103	Digikey		0.01260
Resistor	2	47 Ω	R6, R7	Rohm	RHM47JCT-ND	Digikey		0.01380
Resistor	1	200 kΩ	R5	Rohm	MCR01MRTF2003	Digikey		0.01720
Resistor Array	1	4 x 10 kΩ	RA1	Vishay	CRA04S08310K0JTD	Digikey		0.03220
Capacitor	1	1.5 µF	C1	AVX	TPSA155K025R3000	Digikey	1.13000	0.94000
Capacitor	2	2.2 µF	C3, C5	AVX	TPSA225K035R1500	Digikey	1.01000	0.84600
Capacitor	2	4.7 µF	C7, CA4	Murata	GRM155R60J475M	Digikey	0.38000	0.29500
				Taiyo Yuden	JMK105BBJ475MV-F	Digikey	0.35000	0.16200
Capacitor	1	10 µF	CA2	Vishay	298D106X0004K2T	Digikey	2.14000	1.75500
Capacitor	2	100 nF	C6, CA1	Murata	GRM155R61C104K	Digikey		0.01000
Capacitor	1	220 nF	CA3	Murata	GRM155R60J224K	Digikey		0.05640
Quad Bus Buffer	1		IC1	NXP	N74F126D	Digikey	0.46000	0.34120
Acc. & Compass	1		U5	ST	LSM303DLHC	Digikey	8.20000	6.71000
Volt. Lev. Transl.	1		U27	TI	PCA9306DCUR	Digikey	0.94000	0.74880
LED Green	1		GREEN	Panasonic	LNJ347W83RA	Digikey	0.51000	0.29760
LED Orange	1		ORGNE	Panasonic	LNJ847W86RA	Digikey	0.51000	0.29760
Controller	1		U1	Atmel	ATMEGA1284[P]-AU[R]†	Digikey	7.44000	6.22800
Fuse	1	500 mA	FUSE	Littlefuse	0157.500DR	Digikey	3.18000	2.92200
Power Regulator	1	3.3 V	VR3.3	Linear	LT1761MPS5-3.3	Digikey	4.62000	4.12520

Table continues on next page

Table B.4 continued

Description	Qty/ Module	Value	Parts	Manufacturer	Man Part No	Vendor	Price	Price*
Power Regulator	1	5.0 V	VR5.0	Micrel	MIC5219-5.0YM5 TR	Digikey	1.43000	1.14000
Resonator	1		U7	Murata	CSTCE16M0 V53-R0	Digikey	0.69000	0.49000
FFC Connector	5		U9, U10, U11, U12, U13	TE Conn.	1-84981-0	Digikey	0.19000	0.13240
FFC Conn. (Prog.)	1		PROG	FCI	HFW6R-2STE1LF	Digikey	0.59000	0.36900
N-FET for Sensors	1		SENSFET	Vishay	2N7002K-T1-E3	Digikey	0.38000	0.17240

† ATMEGA1284P-AU, ATMEGA1284P-AUR, ATMEGA1284-AU are functionally identical for this application. Letter P indicates additional power saving features not used in this application, letter R indicates "tape and reel" packaging. During programming of the controller, care must be taken to set the device type to ATMEGA1284/ATMEGA1284P correctly or programming will fail.

* Unit price at order quantity 50.

Table B.5: Bill of Materials, Connector Printed Circuit Board.

Description	Qty/ Module	Value	Parts	Manufacturer	Man Part No	Vendor	Price	Price*
Heater Switch	1	N-FET	U1	Fairchild	FDME410NZT	Digikey	0.700	0.542
High Power Resistor	8	10 Ω, 1 W	R1-8	TE Connectivity	1622820-4	Digikey	0.140	0.083
Alternative Resistor	—	10 Ω, 2 W	—	Bourns	CRM2512-FX10R0ELF	Digikey	0.540	0.430
Alternative Resistor	—	10 Ω, 16 W	—	Susumu	CPA2512Q10R0JNEGHP	Digikey	3.180	2.565
Flat Flex Cable	1			Wavelink	100-010-0020-AC	Wavelink	2.01@500	

* Unit price at order quantity 50.

Table B.6: Bill of Materials, Sensor Printed Circuit Board.

Description	Qty/ Module	Value	Parts	Manufacturer	Man Part No	Vendor	Price	Price*
Proximity Sensor	1		U1	Osram	SFH 7741-Z	Digikey	1.630	1.358
Capacitor	1	1.0 µF	C1	Capacitor	GRM155R60J105ME19D	Digikey	0.100	0.022
Resistor	1	2 kΩ	R2	Resistor	RMCF0402FT2K00	Digikey	0.040	0.013
Resistor	1	10 kΩ	R1	Resistor	MCR01MRTJ103	Digikey	0.040	0.013

* Unit price at order quantity 50.

Table B.7: Bill of Materials, Battery Printed Circuit Board.

Description	Qty/ Module	Value	Parts	Manufacturer	Man Part No	Vendor	Price	Price*
Resistor	1	0.43 Ω	R4	Panasonic	ERJ-2BQFR43X	Digikey	0.59	0.2416
Resistor	1	1 kΩ	R10	Rohm	MCR01MRTJ102	Digikey		0.0126
Resistor	2	2.4 kΩ	R1, R2	Stackpole	RMCF0402JT2K40	Digikey	0.02	0.0092
Resistor	1	20 kΩ	R7	Susumu	RR0510P-203-D	Digikey	0.08	0.0474
Resistor	2	7.5 kΩ	R8, R9	Susumu	RR0510P-752-D	Digikey	0.08	0.0474
Resistor	1	83 kΩ	R5	TE Conn.	A102992CT-ND	Digikey	0.37	0.2570
Resistor Array	1	100 kΩ	RA1	Panasonic	EXB-28V104JX	Digikey	0.20	0.0782
/midrule Capacitor	2	10 µF	C1, C2	Taiyo Yuden	TMK316F106ZL-T	Digikey	0.21	0.0850
Capacitor	3	0.1 µF	C5, C7, C8	Murata	GRM155R61C104KA88D	Digikey		0.0092
Capacitor	1	4.7 µF	C4	TDK	C2012X5R1C475K/1.25	Digikey	0.24	0.1650
Inductor	1	22 µH	L1	Taiyo Yuden	CBC3225T220MR	Digikey	0.23	0.1988
Switching Diode	1		D1	AVX	SD0805S020S1R0	Digikey	0.48	0.4200

Table continues on next page

Description	Qty/ Module	Value	Parts	Manufacturer	Man Part No	Vendor	Price	Price*
LED Blue	1		BLUE	Vishay	VLMB1500-GS08	Digikey	0.62	0.3500
LED Pink	1		PINK	Rohm	SMLP12HBC7W1	Digikey	0.67	0.4320
Battery Charging IC	1		U1	TI	BQ24105RHLR	Digikey	6.97	5.6972
Voltage Comparator	1		U2	TI	LM211DR	Digikey	0.54	0.3972
Battery Spring	1			Keystone	211	Digikey	0.14	0.1134

* Unit price at order quantity 50.

BIBLIOGRAPHY

[1] M. Yim, "A reconfigurable modular robot with many modes of locomotion," in *Proc. of the JSME Intl. Conf. on Advanced Mechatronics*. Tokyo, Japan: JSME, 1993, pp. 283–288.

[2] S. Murata, H. Kurokawa, and S. Kokaji, "Self-assembling machine," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 1. San Diego, CA, USA: IEEE, 1994, pp. 441–448.

[3] M. Yim, D. Duff, and K. D. Roufas, "PolyBot: a modular reconfigurable robot," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 1. San Francisco, USA: IEEE, 2000, pp. 514–520.

[4] M. Yim, Z. Ying, and D. Duff, "Modular robots," *IEEE Spectrum*, vol. 39, no. 2, pp. 30–34, 2002.

[5] M. Yim, W.-m. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, "Modular Self-Reconfigurable Robot Systems [Grand Challenges of Robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.

[6] M. Yim, P. J. White, M. Park, and J. Sastra, "Modular Self-Reconfigurable Robots," in *Encyclopedia of Complexity and Systems Science*. Springer, 2009, pp. 5618–5631.

[7] M. Park, S. Chitta, A. Teichman, and M. Yim, "Automatic Configuration Recognition Methods in Modular Robots," *The International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 403–421, Mar. 2008.

[8] I.-M. Chen and J. W. Burdick, "Enumerating the Non-Isomorphic Assembly Configurations of Modular Robotic Systems," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 702–719, Jul. 1998.

[9] S. C. Goldstein, J. D. Campbell, and T. C. Mowry, "Programmable matter," *Computer*, vol. 38, no. 6, pp. 99–101, 2005.

[10] C. Y. Baldwin and K. B. Clark, *Design rules, Volume 1: The Power of Modularity*. Boston, MA, USA: MIT Press, 2000.

[11] K. Stoy, D. Brandt, and D. J. Christensen, *Self-Reconfigurable Robots: An Introduction*. Cambridge, Mass: MIT Press, 2010.

[12] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Self Organizing Robots Based on Cell Structures - CEBOT," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Tokyo, Japan: IEEE/RSJ, 1988, pp. 145–150.

[13] T. Fukuda and Y. Kawauchi, "Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 1990, pp. 662–667.

[14] G. Beni and J. Wang, "Swarm Intelligence in Cellular Robotic Systems," in *Proc. of the NATO Advanced Workshop on Robots and Biological Systems*. Tuscany, Italy: NATO, 1989.

[15] G. Beni, "From Swarm Intelligence to Swarm Robotics," in *Lecture Notes in Computer Science: Swarm Robotics*. Springer, 2004, ch. 1, pp. 1–9.

[16] A. Castano, R. Chokkalingam, and P. M. Will, "Autonomous and Self-Sufficient CONRO Modules for Reconfigurable Robots," in *Distributed Autonomous Robotic Systems 4*, L. E. Parker, G. Bekey, and J. Barhen, Eds. Springer Japan, 2000, ch. 5, pp. 155–164.

[17] W.-M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh, "Multimode locomotion via SuperBot robots," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Orlando, USA: IEEE, 2006, pp. 2552–2557.

[18] G. G. Ryland and H. H. Cheng, "Design of iMobot, an intelligent reconfigurable mobile robot with novel locomotion," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Anchorage, AK, USA: IEEE, May 2010, pp. 60–65.

[19] P. Moubarak and P. Ben-Tzvi, "Modular and reconfigurable mobile robotics," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1648–1663, Dec. 2012.

[20] V. Zykov, E. Mytilinaios, B. Adams, and H. Lipson, "Self-reproducing machines," *Nature*, vol. 435, no. 7038, pp. 163–164, 2005.

[21] V. Zykov, E. Mytilinaios, H. Lipson, and M. Desnoyer, "Evolved and designed self-reproducing modular robotics," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 308 – 319, 2007.

[22] P. J. White, H. Lipson, and K. Kopanski, "Stochastic self-reconfigurable cellular robotics," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 3. New Orleans, LA, USA: IEEE, 2004, pp. 2888–2893.

[23] P. J. White, V. Zykov, J. Bongard, and H. Lipson, "Three dimensional stochastic reconfiguration of modular robots," in *Robotics: Science and Systems*, 2005, pp. 161–168.

[24] K. Gilpin, K. Kotay, and D. Rus, "Miche: Modular Shape Formation by Self-Dissasembly," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Rome, Italy: IEEE, 2007, pp. 2241–2247.

[25] E. Klavins, "Programmable Self-Assembly," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 43–56, Aug. 2007.

[26] N. Inou, K. Minami, and M. Koseki, "Group Robots Forming a Mechanical Structure-Development of slide motion mechanism and estimation of energy consumption of the structural formation," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Kobe, Japan: IEEE, 2003, pp. 874–879.

[27] R. Fitch and Z. Butler, "Million Module March: Scalable Locomotion for Large Self-Reconfiguring Robots," *International Journal of Robotics Research*, vol. 27, no. 3, p. 12, 2008.

[28] J. D. Hiller and H. Lipson, "Design and analysis of digital materials for physical 3D voxel printing," *Rapid Prototyping Journal*, vol. 15, no. 2, pp. 137–149, 2009.

[29] J. D. Hiller, J. Miller, and H. Lipson, "Microbricks for Three-Dimensional Reconfigurable Modular Microsystems," *Journal of Microelectromechanical Systems*, vol. 20, no. 5, pp. 1089–1097, Oct. 2011.

[30] K. C. Cheung and N. Gershenfeld, "Reversibly assembled cellular composite materials." *Science*, vol. 341, no. 6151, pp. 1219–21, Sep. 2013.

[31] R. MacCurdy, A. McNicoll, and H. Lipson, "Bitblox: A Printable Digital Material for Electromechanical Machines," *International Journal of Robotics Research*, vol. submitted, 2013.

[32] F. Nigl, S. Li, J. E. Blum, and H. Lipson, "Structure-Reconfiguring Robots: Autonomous Truss Reconfiguration and Manipulation," *IEEE Robotics & Automation Magazine*, vol. 20, no. 3, pp. 60–71, Sep. 2013.

[33] R. J. Jackman, S. T. Brittain, A. Adams, M. G. Prentiss, and G. M. Whitesides, "Design and fabrication of topologically complex, three-dimensional microstructures," *Science*, vol. 280, no. 5372, pp. 2089–2091, 1998.

[34] N. Bowden, A. Terfort, J. Carbeck, and G. M. Whitesides, "Self-assembly of mesoscale objects into ordered two-dimensional arrays," *Science*, vol. 276, no. 5310, p. 233, 1997.

[35] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, pp. C47–C52, 1999.

[36] J. von Neumann and A. W. Burks, *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

[37] D. Floreano and C. Mattiussi, *Bio-Inspired Artifical Intelligence*, 1st ed. Boston, MA, USA: MIT Press, 2008.

[38] M. Yim, "Locomotion With A Unit-Modular Reconfigurable Robot," PhD, Stanford Univeristy, 1994.

[39] Schunk Gmbh & Co KG, "Tomorrow starts today! Welcome to SCHUNK – a Pioneer of Modular Robotics," Lauffen, Germany, 2010. [Online]. Available: http://www.schunk.com/schunk_files/attachments/ModularRobotics_2010-06_EN.pdf

[40] G. Beni, "The concept of cellular robotic system," in *Proc. of the Int. Symposium on Intelligent Control*. Arlington, VA, USA: IEEE, 1988, pp. 57–62.

[41] T. Fukuda and S. Nakagawa, "Dynamically reconfigurable robotic system," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Philadelphia, PA: IEEE, 1988, pp. 1581–1586.

[42] P. Miller, "Swarm Theory," *National Geographic Magazine*, Jul. 2007.

[43] C. Zimmer, "From Ants to People, an Instinct to Swarm," Nov. 2007. [Online]. Available: http://www.nytimes.com/2007/11/13/science/13traff.html

[44] E. Sahin, "Swarm Robotics: From Sources of Inspiration to Domains of Application," in *Lecture Notes in Computer Science: Swarm Robotics*. Springer, 2004, pp. 10–20.

[45] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. of the Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*. New York, NY, USA: ACM Press, 1987, pp. 25–34.

[46] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a Colony of Cooperating Agents." *Transactions on Systems, Man, and Cybernetics*, vol. 26, no. 1, pp. 29–41, Jan. 1996.

[47] T. Fukuda, H. Mizoguchi, K. Sekiyama, and F. Arai, "Group behavior control for MARS (micro autonomous robotic system)," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 2. Detroit, MI, USA: IEEE, 1999, pp. 1550–1555.

[48] F. Mondada and E. Franzi, "Biologically inspired mobile robot control algorithms," in *Proc. of the NFP-PFR 23 Symposium*, Zurich, Switzerland, 1993, pp. 47–60.

[49] R. Gross, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous Self-Assembly in Swarm-Bots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1115–1130, Dec. 2006.

[50] M. Yim, B. Shirmohammadi, J. Sastra, M. Park, M. Dugan, and C. Taylor, "Towards robotic self-reassembly after explosion," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. San Diego, CA, USA: IEEE, Oct. 2007, pp. 2767–2772.

[51] J. Davey, N. Kwok, and M. Yim, "Emulating self-reconfigurable robots - design of the SMORES system," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Vilamoura, Protugal: IEEE/RSJ, Oct. 2012, pp. 4464–4469.

[52] K. C. Prevas, C. Unsal, M. O. Efe, and P. K. Khosla, "A hierarchical motion planning strategy for a uniform self-reconfigurable modular robotic system," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Washington DC, USA: IEEE, 2002, pp. 787–792.

[53] D. J. Christensen, "Evolution of shape-changing and self-repairing control for the ATRON self-reconfigurable robot," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Orlando, FL, USA: IEEE, 2006, pp. 2539–2545.

[54] D. Brandt and D. J. Christensen, "A New Meta-Module for Controlling Large Sheets of ATRON Modules," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. San Diego, CA, USA: IEEE, 2007, pp. 2375–2380.

[55] E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa, and S. Kokaji, "Self-reconfigurable modular robots - hardware and software development in AIST," in *Proc. of the Int. Conf. on Intelligent Systems and Signal Processing*, vol. 1. Changsha, China: IEEE, 2003, pp. 339–346.

[56] B. Madhevan and M. Sreekumar, "Structures and Characteristics in Reconfigurable Modular Robots," in *Advances in Reconfigurable Mechanisms and Robots I*, J. S. Dai, M. Zoppi, and X. Kong, Eds. London: Springer London, 2012, pp. 525–534.

[57] T. Fukuda and S. Nakagawa, "Approach to the dynamically reconfigurable robotic system," *Journal of Intelligent and Robotic Systems*, vol. 1, no. 1, pp. 55–72, 1988.

[58] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Structure decision method for self organising robots based on cell structures - CEBOT," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Scottsdale, AZ: IEEE, 1989, pp. 695–700.

[59] T. Fukuda, M. Buss, H. Hosokai, and Y. Kawauchi, "Cell Structured robotic system CEBOT: Control, planning and communication methods," *Robotics and Autonomous Systems*, vol. 7, no. 2-3, pp. 239–248, Aug. 1991.

[60] M. Yim, "New locomotion gaits," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. San Diego, CA USA: IEEE, 1994, pp. 2508–2514.

[61] G. S. Chirikjian, "Kinematics of a metamorphic robotic system," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 1. San Diego, CA, USA: IEEE, 1994, pp. 449–455.

[62] A. Pamecha, C.-J. Chiang, D. Stein, and G. S. Chirikjian, "Design and Implementation of Metamorphic Robots," in *Proc. of the ASME Design Engineering Technical Conference and Computers in Engineering Converence*. Irvine, CA, USA: ASME, 1996.

[63] S. Murata, H. Kurokawa, and S. Kokaji, "Self-Configurable Machine," *Transactions of the Society of Instrument and Control Engineers*, vol. E-1, no. 1, pp. 187–196, Translated from Trans. SICE, Vol.31, No.2, 2001.

[64] E. Yoshida, S. Kokaji, S. Murata, H. Kurokawa, and K. Tomita, "Miniaturized self-reconfigurable system using shape memory alloy," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 3. Kyongju, Korea: IEEE/RSJ, 1999, pp. 1579–1585.

[65] E. Yoshida, S. Murata, and S. Kokaji, "Micro self-reconfigurable robotic system using shape memory alloy," in *Distributed Autonomous Robotic Systems 4*, L. E. P. Barhen, G. Bekey, and Jacob, Eds. Springer, 2000, pp. 145–154.

[66] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji, "A 3-D self-reconfigurable structure," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 1. Leuven, Belgium: IEEE, 1998, pp. 432–439.

[67] P. M. Will, A. Castano, and W.-M. Shen, "Robot modularity for self-reconfiguration," in *Proc. or the Conf. on Sensor Fusion and Decentralized Control in Robotic Systems*. Boston, MA, USA: SPIE, 1999, pp. 236–245.

[68] A. Castano, W.-M. Shen, and P. M. Will, "CONRO: Towards Deployable Robots with Inter-Robots Metamorphic Capabilities," *Autonomous Robots*, vol. 8, no. 3, pp. 309–324–324, 2000.

[69] A. Castano, A. Behar, and P. M. Will, "The Conro modules for reconfigurable robots," *Transactions on Mechatronics*, vol. 7, no. 4, pp. 403–409, Dec. 2002.

[70] B. Salemi and P. M. Will, "Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 700–712, Oct. 2002.

[71] W.-m. Shen, P. Will, and A. Galstyan, "Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms," *Autonomous Robots*, vol. 17, no. 1, pp. 93–105, 2004.

[72] M. Yim, D. Duff, and K. D. Roufas, "Walk on the wild side [modular robot motion]," *IEEE Robotics & Automation Magazine*, vol. 9, no. 4, pp. 49–53, Dec. 2002.

[73] M. Yim and D. Duff, "Closed-chain motion with large mechanical advantage," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 1. Maui, HI, USA: IEEE/RSJ, 2001, pp. 318–323.

[74] Y. Zhang, K. D. Roufas, and M. Yim, "Software architecture for modular self-reconfigurable robots," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 4. Maui, HI, USA: IEEE/RSJ, 2001, pp. 2355–2360.

[75] M. Yim, K. D. Roufas, D. Duff, and C. Eldershaw, "Connecting and disconnecting for chain self-reconfiguration with PolyBot," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 442–451, Dec. 2002.

[76] K. Kotay, D. Rus, M. Vona, and C. McGray, "The self-reconfiguring robotic molecule," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 1. Louven, Belgium: IEEE, 1998, pp. 424–431.

[77] D. Rus and M. Vona, "A physical implementation of the self-reconfiguring crystalline robot," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 2. San Francisco, CA, USA: IEEE, 2000, pp. 1726–1733.

[78] ——, "Crystalline robots: Self-reconfiguration with compressible unit modules," *Autonomous Robots*, vol. 10, no. 1, pp. 107–124, 2001.

[79] D. Rus, Z. Butler, K. Kotay, and M. Vona, "Self-reconfiguring robots," *Communications of the ACM*, vol. 45, no. 3, pp. 39–45, Mar. 2002.

[80] S. Murata, E. Yoshida, K. Tomita, H. Kurokawa, A. Kamimura, and S. Kokaji, "Hardware design of modular robotic system," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 3. Takamatsu, Japan: IEEE, 2000, pp. 2210–2217.

[81] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: self-reconfigurable modular robotic system," *Transactions on Mechatronics*, vol. 7, no. 4, pp. 431–441, Dec. 2002.

[82] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata, "M-TRAN II: Metamorphosis from a four-legged walker to a caterpillar," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 3. Las Vegas, USA: IEEE/RSJ, 2003, pp. 2454–2459.

[83] A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, and S. Kokaji, "Automatic Locomotion Design and Experiments for a Modular Robotic System," *Transactions on Mechatronics*, vol. 10, no. 3, pp. 314–325, Jun. 2005.

[84] Y. Suzuki, N. Inou, H. Kimura, and M. Koseki, "Reconfigurable group robots adaptively transforming a mechanical structure - Crawl motion and adaptive transformation with new algorithms," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Beijing, China: IEEE/RSJ, Oct. 2006, pp. 2200–2205.

[85] M. Koseki, K. Minami, and N. Inou, "Cellular Robots Forming a Mechanical Structure," in *Distributed Autonomous Robotic Systems 6*, R. Alami, R. Chatila, and H. Asama, Eds. Springer, 2007, pp. 139–148.

[86] M. W. Jorgensen, E. H. Ostergaard, and H. H. Lund, "Modular ATRON: modules for a self-reconfigurable robot," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2. IEEE/RSJ, 2004, pp. 2068–2073.

[87] E. H. Ostergaard, K. Kassow, R. Beck, and H. H. Lund, "Design of the ATRON lattice-based self-reconfigurable robot," *Autonomous Robots*, vol. 21, no. 2, pp. 165–183, 2006.

[88] A. Spröwitz, S. Pouya, S. Bonardi, J. Van den Kieboom, R. Moeckel, A. Billard, P. Dillenbourg, and A. J. Ijspeert, "Roombots: Reconfigurable Robots for Adaptive Furniture," *IEEE Computational Intelligence Magazine*, vol. 5, no. 3, pp. 20–32, 2010.

[89] S. Hossain, C. A. Nelson, and P. Dasgupta, "RoGenSiD – A Rotary-plate Genderless Single-sided Docking Mechanism for Modular Self-reconfigurable Robots," in *Proc. of the Int. Design Engineering Technical Conferences and Computers and Information Engineering Conference (IDETC/CIE)*. Portland, OR: ASME, 2013.

[90] ——, "Hardware design and testing of ModRED: A modular self-reconfigurable robot system," in *Advances in Reconfigurable Mechanisms and Robots I*, J. S. Dai, M. Zoppi, and X. Kong, Eds. Tianjin, China: Springer, 2012, ch. Hardware D, pp. 515–523.

[91] J. Baca, S. Hossain, P. Dasgupta, C. A. Nelson, and A. Dutta, "ModRED: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration," *Robotics and Autonomous Systems*, vol. in press, 2013.

[92] P. Dasgupta, J. Baca, S. Hossain, A. Dutta, and C. A. Nelson, "Mechanical design and computational aspects for locomotion and reconfiguration of the ModRED modular robot," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1359–1360.

[93] J. Liedke, R. Matthias, L. Winkler, and H. Worn, "The Collective Self-reconfigurable Modular Organism (CoSMO)," in *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Wollongong, Australia: IEEE, Jul. 2013, pp. 1–6.

[94] V. Zykov, A. Chan, and H. Lipson, "Molecubes: An Open-Source Modular Robotics Kit," in *Self-Reconfigurable Robotics Workshop at the Int. Conf. of Robotics and Intelligent Systems (IROS)*. San Diego, CA, USA: IEEE/RSJ, 2007.

[95] V. Zykov, P. William, N. Lassabe, and H. Lipson, "Molecubes extended:

Diversifying capabilities of open-source modular robotics," in *Proc. of the Self-Reconfigurable Robotics Workshop at the International Conference for Intelligent Robots and Systems*. Nice, France: IEEE, 2008.

[96] J. Sastra, S. Chitta, and M. Yim, "Dynamic Rolling for a Modular Loop Robot," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 758–773, May 2009.

[97] R. F. Garcia, J. D. Hiller, and H. Lipson, "A vacuum-based bonding mechanism for modular robotics," in *Proc. of the ICRA Workshop on Modular Robots, State of the Art*. Anchorage, AK: IEEE, 2010, pp. 57–62.

[98] R. F. Garcia, J. D. Hiller, K. Stoy, and H. Lipson, "A Vacuum-Based Bonding Mechanism for Modular Robotics," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 876–890, 2011.

[99] G. S. Chirikjian, "Metamorphic Hyper-Redundant Manipulators," in *Proc. of the JSME Int. Conf. on Advanced Mechatronics*. Tokyo, Japan: JSME, 1993, pp. 467–472.

[100] E. Yoshida, S. Kokaji, S. Murata, K. Tomita, and H. Kurokawa, "Miniature self-reconfigurable modular machine using shape memory alloy," *Advanced Robotics*, vol. 13, no. 3, pp. 337–338, Jan. 1998.

[101] K. Hosokawa, T. Tsujimori, T. Fujii, H. Kaetsu, H. Asama, Y. Kuroda, and I. Endo, "Self-organizing collective robots with morphogenesis in a vertical plane," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Leuven, Belgium: IEEE, 1998, pp. 2858–2863.

[102] D. Rus and M. Vona, "Self-reconfiguration planning with compressible unit modules," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 1999, pp. 2513–2520.

[103] C. Unsal, H. Kiliccote, and P. K. Khosla, "I(CES)-cubes: a modular self-reconfigurable bipartite robotic system," *Proc. SPIE, Sensor Fusion and Decentralized Control in Robotic Systems II*, vol. 3839, pp. 258–269, Aug. 1999.

[104] J. W. Suh, S. B. Homans, and M. Yim, "Telecubes: mechanical design of a module for self-reconfigurable robotics," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, vol. 4. Washington DC, USA: IEEE, 2002, pp. 4095–4101.

[105] H. Tokashiki, H. Amagai, S. Endo, K. Yamada, and J. Kelly, "Development of a

transformable mobile robot composed of homogeneous gear-type units," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2. Las Vegas, NV, USA: IEEE/RSJ, 2003, pp. 1602–1607.

[106] J. D. Campbell, P. Pillai, and S. C. Goldstein, "The robot is the tether: active, adaptive power routing modular robots with unary inter-robot connectors," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Edmonton, Canada: IEEE/RSJ, 2005, pp. 4108–4115.

[107] J. D. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp, and T. N. Nguyen, "Programmable parts: a demonstration of the grammatical approach to self-organization," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Edmonton, Canada: IEEE/RSJ, 2005, pp. 3684–3691.

[108] P. J. White and M. Yim, "Scalable Modular Self-reconfigurable Robots Using External Actuation," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. San Diego, CA, USA: IEEE/RSJ, 2007, pp. 2773–2778.

[109] H. Kurokawa, K. Tomita, A. Kamimura, and S. Kokaji, "Distributed Self-Reconfiguration of M-TRAN III Modular Robotic System," *International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 373–386, 2008.

[110] C. A. Nelson, K. D. Chu, and P. Dasgupta, "ModRED: a modular self-reconfigurable robot for autonomous extra-terrestrial exploration and discovery," in *Planetary Rovers Workshop, International Conference for Robotics and Automation (ICRA) 2010*. Anchorage, AK, USA: IEEE, 2010.

[111] K. C. Wolfe, M. S. Moses, M. D. Kutzer, and G. S. Chirikjian, "M3Express: A low-cost independently-mobile reconfigurable modular robot," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Saint Paul, MN, USA: IEEE, May 2012, pp. 2704–2710.

[112] J. Romanishin, K. Gilpin, and D. Rus, "M-Blocks: Momentum-Driven, Magnetic Modular Robots," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Tokyo, Japan: IEEE/RSJ, 2013.

[113] W.-m. Shen, J. Bogdanowicz, W. Chun, M. Yim, P. M. Will, M. Sims, S. P. Colombano, D. Kortenkamp, S. Vanderzyl, E. Baumgartner, and G. J. Taylor, "SuperBots: Modular, Multifunctional, Reconfigurable Robotic System for Space Exploration," in *Proc. of the Space Resources Roundtable VII*. League City, TX, USA: Lunar Exploration Analysis Group (LEAG), 2005, p. 96.

[114] R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, and A. J. Ijspeert, "YaMoR and Bluemove–an autonomous modular robot with Bluetooth interface for exploring adaptive locomotion," in *Climbing and Walking Robots*, M. O. Tokhi, G. S. Virk, and M. A. Hossain, Eds. Springer, 2006, pp. 685–692.

[115] M. Park, S. Chitta, and M. Yim, "Isomorphic gait execution in homogeneous modular robots," in *Workshop on Self-Reconfigurable Modular Robots at Robotics Science and Systems (RSS)*, Philadelphia, PA, USA, 2006.

[116] D. Daidie, O. Barbey, A. Guignard, D. Roussy, F. Guenter, A. Ijspeert, and A. Billard, "The DoF-Box project: An educational kit for configurable robots," in *Int. Conf. on Advanced Intelligent Mechatronics*. IEEE/ASME, 2007, pp. 1–6.

[117] A. Lyder, K. Stoy, and R. F. Garcia, "Mechanical Design of Odin, an Extendable Heterogeneous Deformable Modular Robot," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Nice, France: IEEE/RSJ, 2008, pp. 883–888.

[118] G. G. Ryland and H. H. Cheng, "Novel locomotion of iMobot, an intelligent reconfigurable mobile robot," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Anchorage, AK, USA: IEEE, May 2010, pp. 1108–1109.

[119] E. Schweikardt and M. D. Gross, "roBlocks," in *Proc. of the 8th Int. Conf. on Multimodal interfaces (ICMI)*. New York, New York, USA: ACM Press, 2006, p. 72.

[120] E. Schweikardt, "Designing Modular Robots," PhD Proposal, Carnegie Mellon University, 2008. [Online]. Available: http://www.andrew.cmu.edu/user/eschweik/ES Thesis Proposal.pdf

[121] ——, "Personal Communication," 2013.

[122] G. M. Whitesides and M. Boncheva, "Beyond molecules: self-assembly of mesoscopic and macroscopic components." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 8, pp. 4769–74, Apr. 2002.

[123] R. Großand M. Dorigo, "Self-assembly at the macroscopic scale," *Proceedings of the IEEE*, 2008.

[124] M. T. Tolley and H. Lipson, "On-line assembly planning for stochastically

reconfigurable systems," *International Journal of Robotics*, vol. 30, no. 13, pp. 1566–1584, 2011.

[125] V. Zykov and H. Lipson, "Fluidic stochastic modular robotics: Revisiting the system design," in *Proc. of the Robotics Science and Systems Workshop on Self-Reconfigurable Modular Robots*, Philadelphia, PA, USA, 2006.

[126] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, 2006.

[127] H. Lipson and J. B. Pollack, "Automatic design and manufacture of robotic lifeforms." *Nature*, vol. 406, no. 6799, pp. 974–8, Aug. 2000.

[128] M. Nilsson, "Essential properties of connectors for self-reconfiguring modular robots," in *Proc. of the Workshop on Reconfigurable Robots at the Int. Conf. on Robotics and Automation (ICRA)*. Seoul, South Korea: IEEE, 2001, pp. 4071–4076.

[129] W.-m. Shen and P. M. Will, "Docking in self-reconfigurable robots," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 2. Maui, HI, USA: IEEE/RSJ, 2001, pp. 1049–1054.

[130] K. Kotay, "Self-reconfiguring robots: designs, algorithms, and applications," PhD, Dartmouth College, Jan. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1023054

[131] A. Spröwitz, R. Moeckel, M. Vespignani, S. Bonardi, and A. J. Ijspeert, "Roombots: A hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot," *Robotics and Autonomous Systems*, vol. in press, Sep. 2013.

[132] D. Rus and M. Vona, "A basis for self-reconfiguring robots using crystal modules," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2000, pp. 2194–2202.

[133] E. Yoshida, S. Murata, S. Kokaji, A. Kamimura, K. Tomita, and H. Kurokawa, "Get back in shape! A Hardware Prototype Self-Reconfigurable Modular Microrobot that Uses Shape Memory Alloy," *IEEE Robotics & Automation Magazine*, vol. 9, no. 4, pp. 54–60, Dec. 2002.

[134] C. Unsal and P. K. Khosla, "Solutions for 3D self-reconfiguration in a modular robotic system: implementation and motion planning," in *Proc. SPIE 4196,*

Sensor Fusion and Decentralized Control in Robotic Systems III, vol. 4196. Boston, MA, USA: SPIE, 2000, pp. 388–401.

[135] M. Rubenstein, K. Payne, and P. Will, "Docking among independent and autonomous CONRO self-reconfigurable robots," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. New Orleans, LA, USA: IEEE, 2004, pp. 2877–2882.

[136] A. Sproewitz, M. Asadpour, Y. Bourquin, and A. J. Ijspeert, "An active connection mechanism for modular self-reconfigurable robotic systems based on physical latching," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Pasadena,CA, USA: IEEE, May 2008, pp. 3508–3513.

[137] J. Davey, J. Sastra, M. Piccoli, and M. Yim, "ModLock: A manual connector for reconfigurable modular robots," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Vilamoura, Protugal: IEEE/RSJ, Oct. 2012, pp. 3217–3222.

[138] E. Yoshida, S. Murata, S. Kokaji, K. Tomita, and H. Kurokawa, "Micro self-reconfigurable modular robot using shape memory alloy," *Journal of Robotics and Mechatronics*, vol. 13, no. 2, pp. 212–219, 2001.

[139] E. Yoshida, S. Murata, A. Kamimura, and K. Tomita, "A motion planning method for a self-reconfigurable modular robot," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Maui, HI, USA: IEEE/RSJ, 2001, pp. 590 – 597.

[140] J. Liedke and H. Worn, "CoBoLD — A bonding mechanism for modular self-reconfigurable mobile robots," in *Proc. of the Int. Conf. on Robotics and Biomimetics*. IEEE, Dec. 2011, pp. 2025–2030.

[141] L. Hardesty, "Surprisingly simple scheme for self-assembling robots," Boston, MA, USA, Oct. 2013. [Online]. Available: http://web.mit.edu/newsoffice/2013/simple-scheme-for-self-assembling-robots-1004.html

[142] A. N. Knaian, "Electropermanent magnetic connectors and actuators : devices and their application in programmable matter," PhD, Massachusetts Institute of Technology, 2010.

[143] K. Gilpin and D. Rus, "Modular Robot Systems," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 38–55, Sep. 2010.

[144] K. Gilpin, A. Knaian, and D. Rus, "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," in *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. Anchorage, AK, USA: IEEE, May 2010, pp. 2485–2492.

[145] S. Miyashita, M. Hadorn, and P. E. Hotz, "Water Floating Self-assembling Agents," in *Agent and Multi-Agent Systems: Technologies and Applications*, N.-T. Nguyen, A. Grzech, R. J. Howlett, and L. C. Jain, Eds. Springer, 2007, pp. 665–674.

[146] L. Wang and F. Iida, "Physical Connection and Disconnection Control Based on Hot Melt Adhesives," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1397–1409, Aug. 2013.

[147] R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, and A. J. Ijspeert, "Exploring adaptive locomotion with YaMoR, a novel autonomous modular robot with Bluetooth interface," *Industrial Robot: An International Journal*, vol. 33, no. 4, pp. 285–290, 2006.

[148] S. Q. Field, "Personal Communication," 2009.

[149] C. Mauney, "Thermal Considerations for Surface Mount Layouts," 2010. [Online]. Available: http://focus.ti.com/download/trng/docs/seminar/Topic 10 - Thermal Design Consideration for Surface Mount Layouts .pdf

[150] Vishay Beyschlag, "Thermal Management in Surface-Mounted Resistor Applications," Vishay Beyschlag, Tech. Rep., 2011. [Online]. Available: http://www.vishay.com/docs/28844/tmismra.pdf

[151] K. Ferjutz and J. R. Davis, *ASM Handbook: Volume 6: Welding, Brazing, and Soldering*, 10th ed. Materials Park, OH: ASM, 1993.

[152] J. R. J. Amend and H. Lipson, "freeLoader: An open source universal testing machine for high-throughput experimentation," in *Proc. of the Int. Design Engineering Technical Conferences and Computers and Information Engineering Conference (IDETC/CIE)*. Washington, DC, USA: ASME, 2011.

[153] I. R. Schmolka, "Polyoxyethylene-Polyoxypropylene Aqueous Gels," US Patent 19 584 570, 1973.

[154] V. Lenaerts, C. Triqueneaux, M. Quartern, F. Rieg-Falson, and P. Couvreur, "Temperature-dependent rheological behavior of Pluronic F-127 aqueous

solutions," *International Journal of Pharmaceutics*, vol. 39, no. 1-2, pp. 121–127, 1987.

[155] B. Fussnegger, "Poloxamers (2): Lutrol F 127 (Poloxamer 407)." *Excipients & Actives for Pharma*, vol. 40, no. 2, pp. 7–9, Aug. 2000.

[156] C. Chaibundit, N. M. P. S. Ricardo, F. D. M. L. L. Costa, S. G. Yeates, and C. Booth, "Micellization and gelation of mixed copolymers P123 and F127 in aqueous solution." *Langmuir*, vol. 23, no. 18, pp. 9229–36, Aug. 2007.

[157] M. Krishnan, M. T. Tolley, H. Lipson, and D. Erickson, "Hydrodynamically Tunable Affinities for Fluidic Assembly," *Langmuir*, vol. 25, pp. 3769–3774, 2009.

[158] D. Erickson, "Vivo-Fluidics and Programmable Matter," in *Microfluidics Based Microsystems*. Springer, 2010, pp. 553–576.

[159] BASF, "Lutrol F 127: Poloxamers Ph.Eur., Poloxamer USP/NF," Apr. 2008. [Online]. Available: http://www.pharma-ingredients.basf.com/pdf/Statements/Technical Informations/Pharma Solutions/EMP 030738e_Lutrol F 127.pdf

[160] ——, "Pluronic F127 Block Copolymer Surfactant," May 2004.

[161] ——, "Lutrol F 68 Poloxamer 188 Poloxamers Ph.Eur., Poloxamer USP/NF," Apr. 2008.

[162] B. Fussnegger, "Poloxamers (1): Lutrol F 68 (Poloxamer 188)." *Excipients & Actives for Pharma*, vol. 48, no. 4, pp. 5–6, Apr. 1999.

[163] G. Theraulaz and E. Bonabeau, "Coordination in distributed building," *Science*, vol. 269, no. 5224, pp. 686–688, 1995.

[164] B. T. Kirby, B. Aksak, J. D. Campbell, J. F. Hoburg, T. C. Mowry, P. Pillai, and S. C. Goldstein, "A modular robotic system using magnetic force effectors," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. San Diego, CA, USA: IEEE/RSJ, Oct. 2007, pp. 2787–2793.

[165] T. Fukuda, Y. Kawauchi, and M. Buss, "Communication Method of Cellular Robotics CEBOT as a Selforganizing Robotic System," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Tsukuba, Japan: IEEE/RSJ, 1989, pp. 291–296.

[166] S. Revzen, J. Sastra, N. Eckenstien, and M. Yim, "CKBot Platform for the ICRA 2010 Planetary Challenge," in *Proc. of the ICRA Workshop on Modular Robots, State of the Art*. Anchorage, AK, USA: IEEE, 2010, pp. 11–12.

[167] V. Zykov, "Morphological and Behavioral Resilience against Physical Damage for Robotic Systems," Doctoral Dissertation, Cornell University, 2008.

[168] S. Kernbach, O. Scholz, K. Harada, S. Popesku, J. Liedke, H. Raja, W. Liu, F. Caparrelli, J. Jemai, J. Havlik, E. Meister, and P. Levi, "Multi-robot organisms: state of the art," *Proc. of the ICRA Workshop on Modular Robots, State of the Art*, pp. 1–10, 2010.

[169] H. Lipson and V. Zykov, "Experiment design for stochastic three-dimensional reconfiguration of modular robots," in *Proc. of the Self-Reconfigurable Robotics Workshop at the Int. Conf. for Intelligent Robotics and Systems (IROS)*, 2007.

[170] B. Salemi, M. Moll, and W.-m. Shen, "SUPERBOT: A Deployable, Multi-Functional, and Modular Self-Reconfigurable Robotic System," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. Beijing, China: IEEE/RSJ, Oct. 2006, pp. 3636–3641.

[171] R. F. Garcia, K. Stoy, D. J. Christensen, and A. Lyde, "A self-reconfigurable communication network for modular robots," in *Proc. of the Int. Conf. on Robot Communication and Coordination (ROBOCOMM)*. Piscataway, NJ, USA: IEEE, 2007, pp. 1–8.

[172] V. Kuo and R. Fitch, "Towards a parallel wireless radio communication architecture for modular robots," in *ARAA Australasian Conference on Robotics and Automation*, 2009, pp. 1947–1952.

[173] ——, "A Parallel Wireless Radio Communication Architecture for Modular Robots," in *Proc. of the ICRA Workshop on Modular Robots, State of the Art*. Anchorage, AK, USA: IEEE, 2010, pp. 69–76.

[174] E. Klavins, R. Ghrist, and D. Lipsky, "A grammatical approach to self-organizing robotic systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 949–962, 2006.

[175] P. J. White and M. Yim, "Reliable External Actuation for Full Reachability in Robotic Modular Self-reconfiguration," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 598–612, Nov. 2009.

[176] M. T. Tolley and H. Lipson, "Programmable 3D Stochastic Fluidic Assembly of cm-scale modules," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*. San Francisco, CA, USA: IEEE, Sep. 2011, pp. 4366–4371.

[177] E. Klavins, S. Burden, and N. Napp, "Optimal rules for programmed stochastic self-assembly," in *Robotics: Science and Systems*, Cambridge, MA, 2006, pp. 9–16.

[178] S. Mitri, S. Wischmann, D. Floreano, and L. Keller, "Using Robots to Understand Social Behaviour." *Biological Reviews of the Cambridge Philosophical Society*, vol. 88, no. 1, pp. 31–9, Feb. 2013.

[179] T. Belpaeme and A. Birk, "A multiagent system based on heterogeneous robots," in *Lecture Notes in Computer Science Volume 1456*. Springer, 1998, pp. 13–24.

[180] Google Inc, "What are Chrome Apps?" 2013. [Online]. Available: http://developer.chrome.com/apps/

[181] ——, "AngularJS — Superheroic JavaScript MVW Framework," 2013. [Online]. Available: http://www.angularjs.org/

[182] M. Otto, J. Thornton, and The Contributors to Twitter Bootstrap, "Bootstrap," 2013. [Online]. Available: http://getbootstrap.com/2.3.2/

[183] Electricimp, "specification: imp001," Mountain View, CA, USA, 2013. [Online]. Available: http://electricimp.com/docs/attachments/hardware/datasheets/imp001.pdf

[184] E. Haasdijk, A. A. Rusu, and A. E. Eiben, "HyperNEAT for Locomotion Control in Modular Robots," in *Evolvable Systems: From Biology to Hardware – Proc. of the 9th Int. Conf. on Evolvable Systems*, G. Tempesti, A. M. Tyrrell, and J. F. Miller, Eds. York, UK: Springer, 2010, pp. 169–180.

[185] M. Schmidt and H. Lipson, "Age-fitness pareto optimization," *Genetic Programming Theory and Practice VIII*, vol. 8, pp. 129–146, 2011.

[186] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies." *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, Jan. 2002.

[187] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction

of development," *Genetic Programming and Evolvable Machines*, vol. 8, no. 2, pp. 131–162, May 2007.